

Mechanisms of Planning and Problem Solving in the Brain

JAMES S. ALBUS

*Center for Mechanical Engineering and Process Technology,
National Engineering Laboratory,
National Bureau of Standards, Washington, D.C. 20234*

Received 19 February 1978; revised 31 January 1979

ABSTRACT

Classical AND/OR goal, or task, decomposition techniques are generalized to deal with the problem of sensory-interactive goal-directed behavior in biological organisms. A neurophysiological model is described which demonstrates the capacity to learn, to generalize, to compute multivariate mathematical functions, and to decompose input commands into sequences of output commands in a context-sensitive manner. Evidence is presented that clusters of neurons with such properties are arranged in hierarchical structures in the brain so as to produce AND/OR task compositions. At the lowest levels in the motor system these clusters transform coordinates and compute servo functions. At the middle levels they decompose input commands into sequences of output commands which give rise to behavior patterns. Mechanisms by which feedback can alter these decomposition sequences to compensate for perturbations and uncertainties in the environment are described. At the highest levels of the hierarchy there are goal selecting and evaluating mechanisms. It is argued that in higher mammals these upper levels of the motor hierarchy are the mechanisms of planning and problem solving.

PREFACE

The following paper attempts as far as possible to present a notation which is both intuitively clear and mathematically precise, and to offer supporting evidence wherever possible. The reader should understand, however, that any model of neuronal mechanisms of the higher cognitive processes must of necessity involve speculation and metaphorical language. It should also be understood that a multidisciplinary approach to such a large and complex subject cannot avoid a mixing of jargon and an over simplification of many difficult issues.

1. INTRODUCTION

Planning and problem solving are fundamental components of intelligent behavior in humans, animals, or machines. Being able to contemplate the potential costs and benefits of future actions, to imagine goals, and to plan

sequences of actions directed toward the attainment of those goals may in fact be taken as (at least in part) a definition of intelligence.

It is for this reason that computer-science research in artificial intelligence has historically focused on techniques for problem solving and planning [37]. These methods typically involve a heuristically guided, trial-and-error search through a space of possible solutions [38].

One of the fundamental techniques used in problem solving is the decomposition of problems into subproblems which are simpler to solve than the original problem. This procedure can be repeatedly applied to subproblems, and then to sub-subproblems, until finally the end product is a set of primitive problems for which there are known one-step solutions.

The concept of solving difficult problems, and the closely related concept of controlling large systems, by problem or task decomposition are old ones. It is implicit in many forms of control hierarchies such as exist in military command structures, business management organizations, and industrial manufacturing procedures, and has for more than a century been assumed to be a mechanism used by the brain for generating and controlling behavior [25].

The mathematical notation of AND/OR trees or graphs which can be used to conveniently represent the process of problem decomposition was first introduced by Slagle [55] in his work on symbolic integration. It was applied to the analysis of serial action work in an industrial environment by Rigney and Towne [50]. Today it still forms the principal theoretical framework for the most advanced and successful work in means-ends analysis, situation action rules, and production based problem solvers [37, 70].

The techniques of problem reduction have also been applied to robot control in planning and goal-seeking programs such as were developed for the Stanford Research Institute robot SHAKEY [49, 17, 23]. Similar techniques have recently been employed in a hierarchical planning and execution program called NOAH for robot assembly and disassembly of mechanical equipment [52] as well as in several psychological models [36, 37].

Most artificial-intelligence research in goal or task decomposition has been done in a planning context as opposed to a control context. In planning the emphasis naturally tends toward mechanisms for searching AND/OR graphs to find, and hopefully optimize, solutions. This implies a detached evaluation of alternatives where the constraints of real-time interaction with a dynamic and unpredictable environment can be deemphasized or ignored altogether. Such an approach is quite adequate for dealing with problems in abstract mathematics or board games where response time is not critical and the resulting physical activity is trivial or can be simulated in software. However, there has been a tendency to employ essentially the same methodology in robot control programs such as those devised for SHAKEY and the more recent work in robot assembly without adding

sophisticated sensory interactive mechanisms to translate high-level plans into low-level actions. The effect is to place enormous demands on the planning programs for attending to details, and the resulting behavior appears most unintelligent because of lengthy interplan periods of open-loop activity in which there is little or no interaction with sensory feedback.

2. PLANNING AND GOAL-SEEKING IN THE BRAIN

An obvious but seldom recognized fact is that planning is not characteristic of the behavior of most biological organisms. In most creatures the central nervous system is primarily a control mechanism for goal-seeking, not planning. Only in the most advanced species does the brain demonstrate any significant capacity for foresight, imagination, and systematic evaluation of potential future scenarios in the formulation of plans.

The fact that AND/OR trees do not explicitly represent the time variable tends to blur the important differences between planning and goal-seeking. The distinguishing feature of goal-seeking, as opposed to planning, is that it is a real-time control process resulting in physical activity. Goal-seeking produces a sequence of overt actions which may be represented by a single uninterrupted string of primitive actions, which are terminal symbols of an AND/OR tree.

Planning, on the other hand, is not a real-time control process, nor is it best characterized by a single uninterrupted trajectory. Planning involves the postulation of hypothetical situations, the evaluation of predicted or imagined results, and the optimization of solution paths prior to the initiation of overt behavioral activity. Planning is a non-real-time heuristic search over a space of potential strategies. The benefit of planning is, of course, that it significantly enhances the first-try success probability of goal-seeking behavior by preselecting optimum goal and subgoal decompositions from a large space of possibilities.

The overwhelming weight of evidence from the evolutionary record [26, 48, 67] as well as from contemporary behavioral science [33] indicates that the original and still primary function of the brain is not to think and plan, but to act and react. When one observes the behavior of creatures in the lower to middle ranges of intelligence such as ants, bees, fish, birds, and mice, there is ample evidence for goal-seeking [61] but virtually none for planning. Even the mid to upper reaches of the intelligence scale such as squirrels, dogs, cats, baboons, and apes, one sees very few instances of planning or problem solving per se. The approach to a problem is most often to simply do something, and if that doesn't work, to do something else. Trial-and-error goal-seeking, or searching, of this kind contains the seeds of planning in that it results in a heuristic search over a space of possible solutions that can be described by an AND/OR tree. But it is distinct

from planning in that it involves overt action at every point in the procedure. Hypothetical situations are not postulated, imaginary results are not evaluated, and solution paths are not optimized prior to initiation of behavioral activity.

The ability of many lower animals to pursue long-range goals of great complexity such as migratory behavior, or elaborate sequences of nesting, mating, and young-rearing behavior, should not be confused with planning. In all but the most advanced species, these processes appear to be almost entirely stored algorithmic procedures which are triggered into execution by immediate environmental stimuli interacting with hormonal and appetitive variables [6]. These may affect the selection of goals and strategies but do not alter the immediacy of response. There is no evidence for a search of strategy space prior to the execution of motor activity.

The ability to learn, or modify behavioral algorithms through experience, also has nothing to do with the ability to plan. Virtually all species can learn, or modify their responses to sensory stimuli, but only the most advanced use learned associates to any significant degree for generating internal representations of future or hypothetical situations which can then be evaluated in the formulation of plans.

Of course, there do exist species for which there exists clear evidence of conscious forethought and cost-benefit analysis of contemplated future action. In humans as well as in higher mammals such as porpoises, whales, and apes, and even occasionally in dogs and cats, at least the rudiments of purpose and forethought have been reliably reported [28]. Thus in a few cases, in the last paragraphs of evolutionary history, there has emerged the ability to plan ahead and consciously select among alternative strategies [43].

The rarity and late arrival of the ability to plan suggests that a highly developed precursor, or substrate, was required from which planning capabilities could evolve. Both the similarities and the differences between planning and goal-seeking suggest that the mechanism for sensory-interactive goal-directed behavior may have been this precursor.

The implication is that a sensory-interactive goal-directed motor system is not simply an appendage to the intellect, but is rather the substrate in which intelligence evolved. There is, in fact, no evidence for a clear demarcation between the motor system and the intellect. Quite to the contrary, much anatomical, neurophysiological, and behavioral evidence suggests that complex behavior is generated in a multilevel control hierarchy where motor outputs are merely the terminal symbols of a behavioral AND/OR tree. This is a concept which has been put forward in various forms by Pribram [46], Arbib [6], Greene [21], Tinbergen [61], and many others. Figure 6 below illustrates Tinbergen's hypothesis that the varying degrees of complexity of muscle contractions, motor responses, and

behavioral patterns result from underlying mechanisms consisting of motor centers "organized in a hierarchical system like the staff organization of an army" [61]. Sensory data interact with all levels of the motor hierarchy to select goals, modify task decompositions, and generate behavioral sequences.

Only if the motor system is very narrowly defined as the last few levels in the goal-seeking control hierarchy can motor control be divorced from the higher functions of goal selection, strategy modification, and task decomposition. The implication is that the motor system is much more intimately involved with intelligence than has been generally thought. If so, then an understanding of the human capacities to think and plan may be greatly enhanced by knowledge of how the motor system generates and controls behavior.

3. AND/OR TRAJECTORIES

In order to use classical AND/OR graph theory in dealing with real-time control in a neurophysiological motor hierarchy two basic adaptations are needed:

first, the discrete nature of AND/OR graphs needs to be generalized to cope with the continuous requirements of real-time control, and

second, a neurophysiological mechanism needs to be proposed which can implement the task decomposition operators.

AND/OR problem reduction analysis, like most artificial intelligence graph generating and searching techniques, was developed in the context of list-processing languages such as LISP where nodes correspond to discrete memory locations and edges are presented by discrete pointers. In order to generalize this theory for continuous control problems a notation is required which can merge sequences of AND nodes into smooth trajectories and let alternative OR nodes be capable of lying anywhere within large regions of multidimensional space.

In order to visualize this concept, let us define a task decomposition operator as a continuous single-valued function H which transforms an input vector $\mathbf{S} = (s_1, s_2, \dots, s_N)$ composed of continuous variables s_i into an output vector $\mathbf{P} = (p_1, p_2, \dots, p_L)$ composed of continuous variables p_j . H thus maps each input vector \mathbf{S} in input space into an output vector \mathbf{P} in output space. If the input vector moves as a function of time (as a result of any of its components changing with time) then \mathbf{S} will describe a trajectory T_s through input space. Assume the function H samples the input periodically and after a short computation delay produces an output. Thus as \mathbf{S} moves along T_s , \mathbf{P} will move along T_p . This is illustrated in Fig. 1.

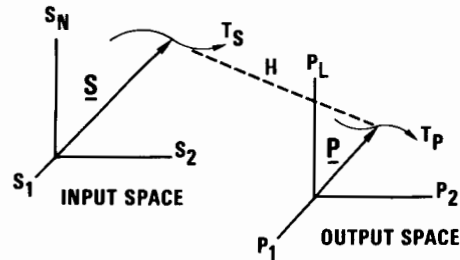


FIG. 1. The function H maps each point in input space into a point in output space. Thus each point on the trajectory T_s is mapped into a point on the trajectory T_p .

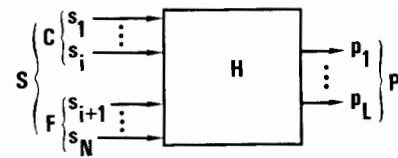


FIG. 2. The input vector S can be divided into a command vector C and a feedback vector F such that $S=C+F$. The output vector is a function of the input such that $P=H(S)$.

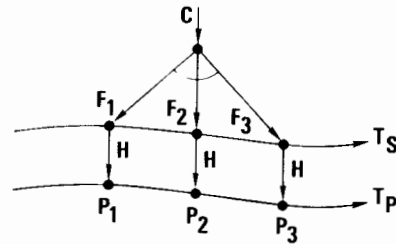


FIG. 3. A continuous AND/OR decomposition where C selects an OR node and F traces out an AND trajectory.

We can now divide the input vector into two parts,

$$C = (s_1, \dots, s_i, 0, \dots, 0)$$

and

$$F = (0, \dots, 0, s_{i+1}, \dots, s_N),$$

such that $S=C+F$ as shown in Fig. 2.

Assume for the moment that C is held constant while F is allowed to vary. If F varies smoothly, the vector S will trace out the trajectory T_s as shown in Fig. 3. The function H maps each point on T_s into a point on T_p .

The tip of the vector C now corresponds to the selection of an OR node. The tip of the vector F traces out a series of AND nodes under this OR node. In the case where F varies smoothly the AND nodes merge into a continuous trajectory. If F moves in steps from F_1 to F_2 to F_3 , then the vector S jumps from one discrete point along T_s to another. Thus the continuous analysis degenerates to the discrete case when the inputs are discrete.

A different value of C defines a different generalized OR node such that as F drives the sequence of subtasks, a different trajectory and hence a different sequence of subtasks is selected, as in Fig. 4.

It is now possible to construct a hierarchy of task decomposition operators such that the output from the highest level becomes input to the second highest level and so on down to the lowest level where the output is a string of terminal symbols, or a trajectory of time-dependent variables which drive physical actuators. Thus is illustrated in Fig. 5. Here the input command to the highest level is a variable list which defines a complex task ASSEMBLE AB. This complex task combined with feedback forms an input vector which is transformed by the highest-level H operator into a simple task command FETCH A. As the highest-level feedback changes, the output vector moves along a trajectory, creating a sequence of simple tasks FETCH B, MATE B TO A, FASTEN B TO A, etc. The second H operator accepts these commands and generates a sequence of elemental moves REACH TO A, GRASP, MOVE TO C, RELEASE, etc. as its feedback changes. The third-level H operator accepts each elemental move combined with feedback and produces a sequence of velocity commands in x, y, z coordinates. At the bottom level the H operator uses feedback to transform from cartesian coordinates into rotary-joint angle coordinates to drive the actuators.

Feedback enters this hierarchy at every level. At the lowest levels this feedback may simply be unprocessed position, velocity, or force information required for servo control. At higher levels this feedback may have passed through several stages of feature extraction and pattern recognition in order to detect the completion of subtasks or errors in subtask perfor-

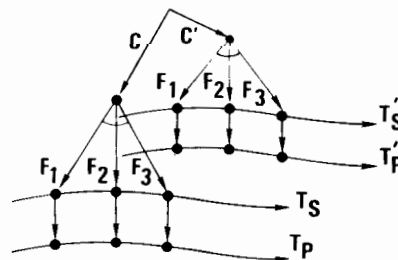


FIG. 4. Two different commands C and C' select two different OR nodes.

mance. This allows the decomposition operator to step to the next subtask in sequence or to an alternative subtask for error correction. The feedback thus closes a real-time control loop at each level in the hierarchy.

The F vector (or at least every element in the F vector) need not originate from the environment in every case. If the sequence of subtasks is fixed, i.e. not dependent on the environment, then F may simply convey timing information to drive P along a predetermined trajectory. Elements in F may also derive from the P vector at the same or a lower level of the hierarchy, or from another hierarchy in a different sensory-motor system.

Note that Fig. 5 has only AND nodes at each level. Only the trajectories of the P vectors at each level were shown, in order to simplify the figure. Note also that the time axis is explicitly represented. This means that the nodes of Fig. 5 are analogous to world points (in relativity theory [29]) and the trajectories analogous to world lines. Each point on a trajectory corresponds to the value of a P vector at a moment in time. The C and F vectors which define the trajectories in Fig. 5 exist in a space which is orthogonal to the time axis. Thus, for example, the C_3 command *FETCH A* combines with the initial F_3 feedback to produce the initial P_3 output *REACH TO A*. As F_3 changes the output P_3 moves to *GRASP*, then to *MOVE TO C*, and then to *RELEASE*. The explicit representation of time stretches the trajectory T_{P_3} out along the horizontal axis so as to produce a left-to-right ordering.

Alternative trajectories which might have been selected under a different set of command and feedback vectors are not shown. To illustrate all of the trajectories which might have occurred over the time interval of Fig. 5 would clearly be impossible.

It is possible, however, at a single instant of time to illustrate the entire set of alternative P vectors which might have occurred. Figure 6 illustrates a snapshot of the feedback-dependent selection of alternative P vectors in the behavior-generating hierarchy of the male three-spined stickleback fish [61]. Figure 6 represents a single cut through world space orthogonal to the time axis. Figure 7 suggests how as time progresses P vectors produced at each instant combine to trace out trajectories.

The discreteness of the nodes in Figure 6 is, of course, an oversimplification. Although the processing of sensory feedback through several layers of pattern recognition operators tends to make potential P vectors at the higher levels cluster in reasonably compact regions of space, nevertheless, these regions are by no means points. Tinbergen reports variations in the "intensity" of behavioral patterns such as *THREATEN* or *DANCE* as well as in the "intensity" of the feedback stimulus required to elicit particular patterns [61]. At all levels variations in F vectors due to irregularities in the environment cause all the nodes selected to vary more or less from one task performance to the next. Such variations cause trajectories such as are shown in Figs. 5-7 to differ from one task performance to the next. This

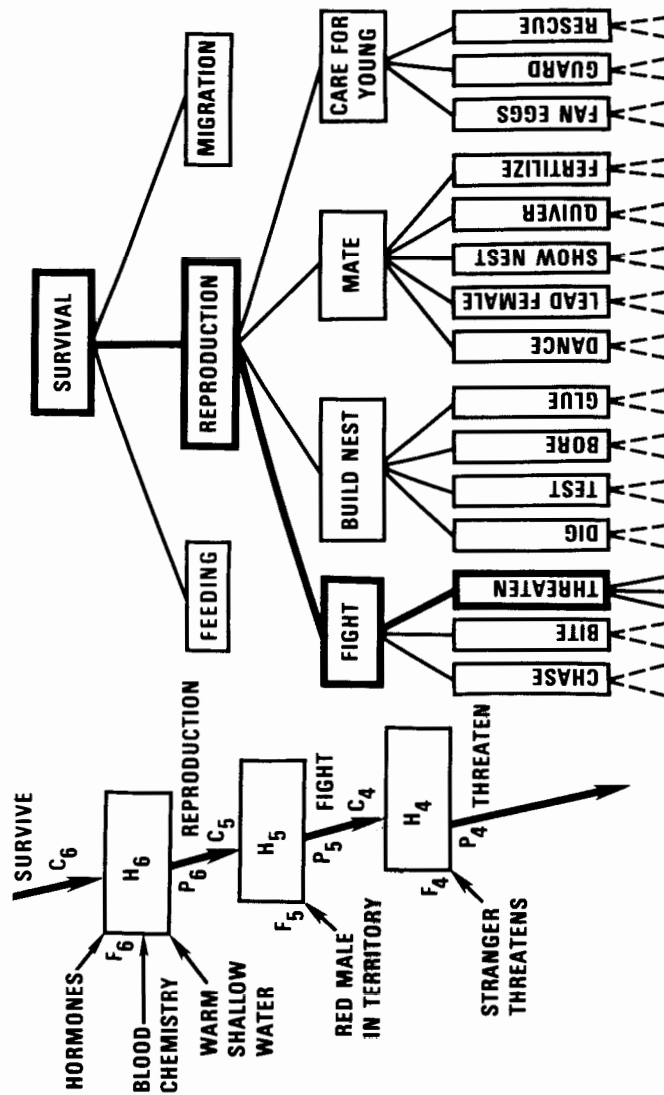


FIG. 6. The upper levels of a hierarchy of goal-selecting operators, illustrating the generation of instinctive behavior in the male three-spined stickleback. After Tinbergen [61].

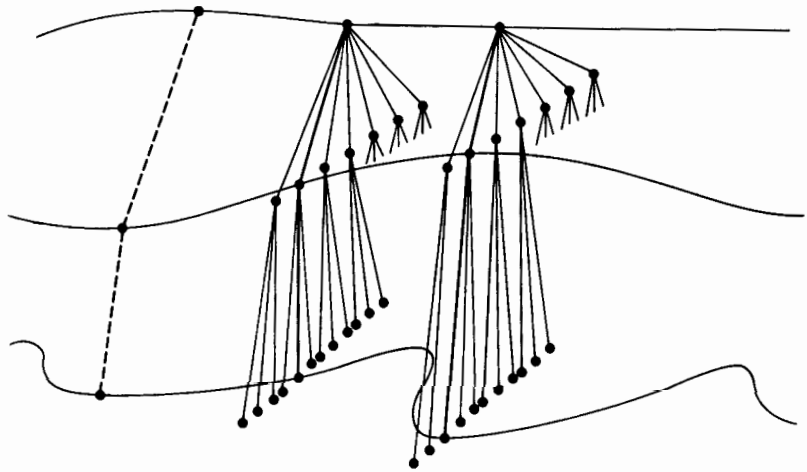


FIG. 7. As time progresses the P vectors selected at each instant flow together to generate trajectories.

implies that while there may exist an ideal trajectory through S and P space at each level of the hierarchy corresponding to an ideal, or perfect, task performance, there also must be an envelope of close-to-ideal trajectories at each level which correspond to successful task performance. Furthermore, the H functions must be such as to provide restoring forces which compensate for perturbances and so keep the vectors corresponding to any particular instantiation of a task performance within an envelope of successful trajectories.

Small perturbations can be corrected by low-level feedback loops as in Fig. 8. These involve very little processing of the feedback data and hence are fast acting.

Large perturbations in the environment may require strategy changes at higher levels in order to accomplish the top-level input command, as illustrated in Fig. 9. Major changes in the environment are detected at higher levels after processing through several levels of pattern recognizers. This produces differences in the F vector at the higher level, which in turn produces different C vectors to the lower levels. Again assuming the proper H function, this produces an alternative high-level strategy to cope with the perturbation. Of course, if the H functions do not provide stability, or if the environment is so perverse that the system is overwhelmed, then the trajectories diverge from the region of successful performance and failure occurs.

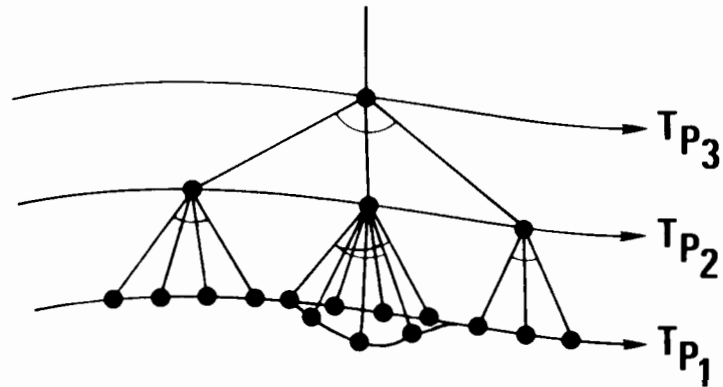


FIG. 8. A small perturbation in the trajectory T_{P_1} can be corrected by a low-level H function.

Overlearned tasks correspond to those for which the H functions at the lower levels are sufficiently well defined over a large enough region of their input space so as to maintain the terminal trajectory well within regions of stability and success without requiring intervention by the higher levels for strategy modification. In such cases, the upper levels of the hierarchy need only select and issue the proper mid-level task command. The remainder of the hierarchy executes this command using only low-level feedback. Only when unusual environmental conditions arise is additional processing required by the higher levels.

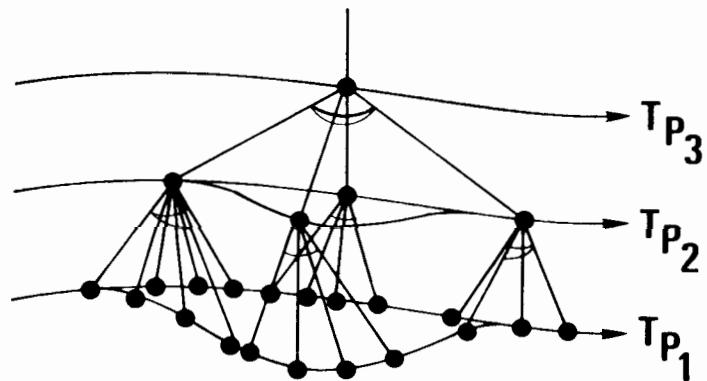


FIG. 9. A large perturbation in the trajectory T_{P_1} may require a strategy change at higher level. The change in T_{P_2} represents the selection of a different OR trajectory.

This suggests how the mechanisms of thinking and planning could evolve in the upper levels of a multilevel hierarchy. Once the hierarchy is sufficiently complex and the lower-level operation sufficiently well learned, then the upper levels need not be continuously involved in supervising the execution of routine tasks. They may instead be diverted into other uses such as planning, problem solving, or simply daydreaming. This might explain how it is possible for a person to get into a car in the morning and drive to work with hardly a second thought given to the details of driving. These details are adequately handled by the lower-level feedback loops. The upper levels of the motor hierarchy are then free to attend to other things.

It is evident that the properties of the H function and C and F vectors are different at different levels of the hierarchy. At the lowest levels H may implement a set of coordinate transformations or servo computations. In this case C and F are lists of arguments and P is a vector of computed solutions. At another level, H may implement a task decomposition operator which breaks tasks into a sequence of subtasks. In this case C may be interpreted as a symbolic command plus a variable list which defines a procedure. The feedback vector F carries information from the environment, which in combination with C determines the particular output which the procedure produces. Each output P is itself a list of symbols defining a procedure for the next lower level. At the upper levels, the H operator has the characteristics of an IF/THEN production rule. The S vector is the premise IF, the P vector the THEN output. Recent work in artificial intelligence [10] has shown that IF/THEN production rules are a very convenient and successful formalism for representing knowledge, solving problems, making plans, and generating English-language translations of logical inference.

The important feature of the H operator is that despite differences in the function embodied at the different levels in the hierarchy, the basic structure of the H operator remains the same. The differences result from different characteristics of the input C and F vectors and different transfer functions in H . This suggests that the same type of anatomical components which are used by the brain in the lower and mid levels of the control hierarchy to produce sensory interactive motor behavior may, with only minor modifications, be used at the upper levels of the same hierarchy to plan and solve problems.

The implication again is that the intellect is merely an extrapolation of the motor system, i.e., that it uses the same basic components arranged in the same basic structure. In this analysis the phenomena of memory, of conditioning, of instinct, of discrimination learning are not separate from the sensory-motor system but are embedded in the H functions which compose the sensory-motor system.

4. A NEUROPHYSIOLOGICAL MODEL

The foregoing discussion is based almost entirely on concepts drawn from artificial-intelligence research, physiological and behavioral psychology, and control theory. These arguments would be much more satisfying if there were some solid evidence that structures capable of producing such performance actually existed in the brain.

It was a elegant series of experiments during the 1960s by Eccles, Ito, and Szentagothai [12] which produced the evidence for a mathematical model of the cerebellum which was independently developed in Great Britain by David Marr [34] and in the United States by the author [3]. Marr published his model first in 1969, and since then it has become a widely *accepted working hypothesis among cerebellar neurophysiologists* [8]. The essential outline of this model is shown in Fig. 10.

Further work on this model has led to a mathematical formalism which has all the properties of the H functions described in the preceding section. This formalism, called the cerebellar model arithmetic computer (CMAC) [1,2,4], can be described as a computing device which accepts an input

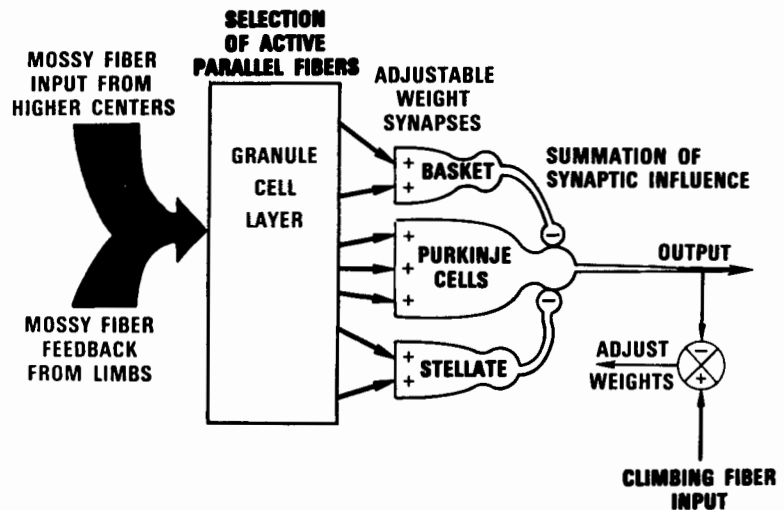


FIG. 10. A model of the cerebellar cortex. Commands and feedback both enter via mossy fibers. The granular layer maps mossy fiber input vectors into subsets of active parallel fibers. Adjustable synaptic weights produce a discriminate function on the parallel fibers. The Purkinje sums positive inputs directly and negative inputs via basket and stellate cells. Weight adjustment training is controlled by the climbing fiber input.

vector $\mathbf{S}=(s_1, s_2, \dots, s_N)$ and produces an output value $p=h(\mathbf{S})$. L of these devices operating on the same input vector produces an output vector $\mathbf{P}=H(\mathbf{S})$.

Each CMAC may be represented as a series of mappings

$$S \rightarrow M \rightarrow A \rightarrow P,$$

where

$$S = \{\text{input vectors}\},$$

$$M = \{\text{intermediate variables}\},$$

$$A = \{\text{association cell vectors}\},$$

$$P = \{\text{output vectors}\}.$$

CMAC is similar in overall form to the perceptron [51,3], but the special character of the $S \rightarrow M$ mapping and the nonbinary input and output capability give it computational powers which far exceed those of the perceptron family of learning machines, or similar devices such as maximum-likelihood classifiers and threshold logic units [39].

4.1. THE $S \rightarrow M$ MAPPING

Each input vector $\mathbf{S}=(s_1, s_2, \dots, s_N)$ is composed of N variables, which may be either continuous or discrete.

The $S \rightarrow M$ mapping is a series of mappings, one for each input variable:

$$S \rightarrow M = \left\{ \begin{array}{l} s_1 \rightarrow m_1^* \\ s_2 \rightarrow m_2^* \\ \vdots \\ s_N \rightarrow m_N^* \end{array} \right\}.$$

The range of each s_i is coarsely quantized by K quantizing functions C_1, C_2, \dots, C_K , each of which is offset by one K th of the quantization interval. m_i^* is the set of values given by the K quantizing functions.

A simple example of this mapping is illustrated in Fig. 11. Here the two input variables s_1 and s_2 are represented with unity resolution on the range 0 to 16. The range of each input variable is also covered by four intermediate

variables or quantizing functions of lower resolution. In Figure 11, s_1 is mapped into a set m_1 composed of four intermediate variables:

$$m_1 = \{C_1, C_2, C_3, C_4\},$$

where

$$C_1 = \{A, B, C, D, E\},$$

$$C_2 = \{F, G, H, J, K\},$$

$$C_3 = \{M, N, P, Q, R\},$$

$$C_4 = \{S, T, V, W, X\}.$$

For every value of s_1 , there exists a unique set of elements m_1^* , one from each set of intermediate variables in m_1 , such that the value of s_1 uniquely defines the set m_1^* and vice versa. For example, in Fig. 11 the value $s_1 = 7$ maps into the set $m_1^* = \{B, H, P, V\}$ and vice versa. Similarly, the value $s_2 = 10$ maps into the set $m_2^* = \{c, j, q, v\}$ and vice versa.

The $S \rightarrow M$ mapping corresponds to that accomplished by sensory end-organs in biological systems. In the body, the angular position of a joint, the tension in a tendon, the velocity of contraction of a muscle are all precise physical parameters analogous to the variables s_i . Each of these is encoded by a multiplicity of sensory organs into firing rates on neuron axons, which are relatively imprecise information channels. m_i^* corresponds to the set of nerve fibers which are maximally stimulated by the instantiation of the variable s_i .

One result of the $S \rightarrow M$ transformation is that a single precise variable s_i can be transmitted over a multiplicity of imprecise information channels such as neuron axons. A second, equally important result is that small changes in the value of the input variable s_i have no effect on most of the elements in m_i^* . This leads to the property of generalization, which is so important to learning and recall in a world where no two situations are ever exactly the same.

4.2. THE $M \rightarrow A$ MAPPING

In the cerebellum, incoming nerve fibers enter what is known as the granular layer, where they make contact with a set of association cells, called granule cells. A system of negative feedback similar to an automatic

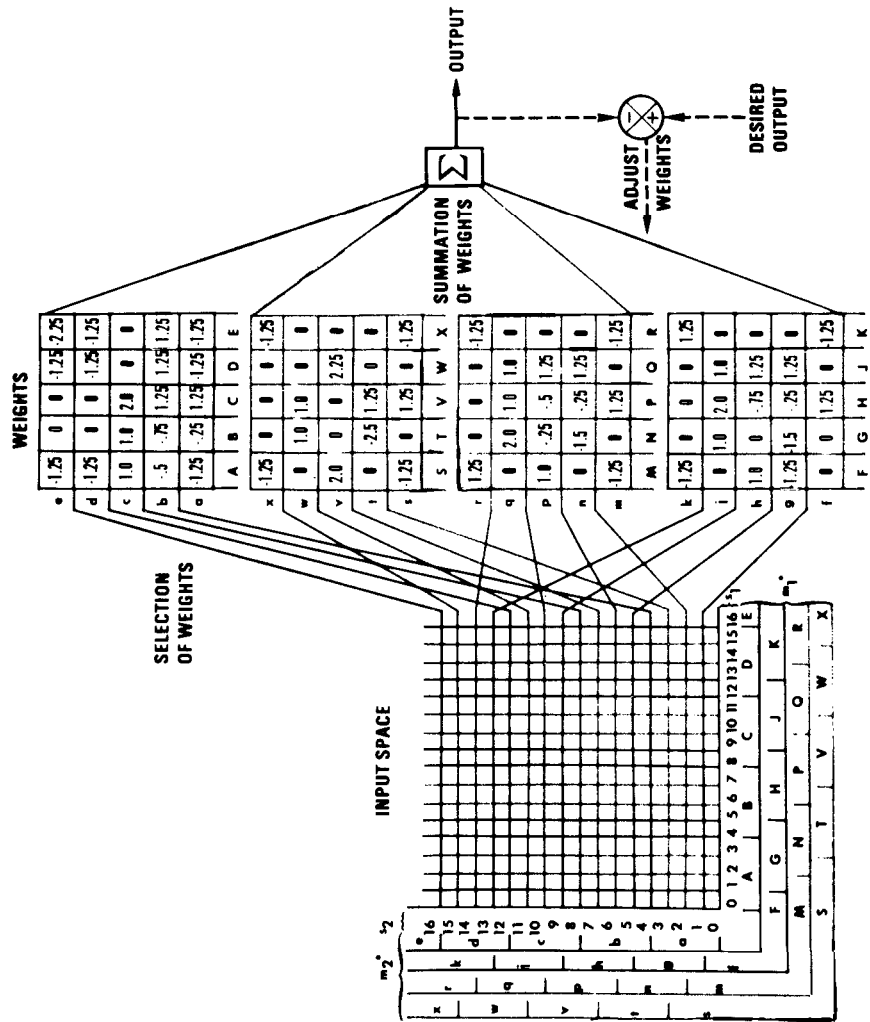


FIG. 11. CMAC maps each input variable into a set of intermediate variables (in this illustration a set of four). These sets of intermediate variables are then combined to select a set of weights. The sum of the weights is the CMAC output. Refer to text for detailed explanation.

gain control regulates the overall activity in the granular layer so that a small and fairly constant percentage of the granule cells are stimulated into an active state. This is simulated in CMAC by the concatenation of respective elements in m_i^* to select a set of nonzero elements A^* out of the association cell vector A . For example, in Fig. 11 the sets $m_1^* = \{B, H, P, V\}$ and $m_2^* = \{c, j, q, v\}$ are combined to select the set of elements $A^* = \{Bc, Hj, Pq, Vv\}$. The number of elements in A^* is thus equal to K , the number of quantizing functions chosen for the $S \rightarrow M$ mapping.

Note that the element Bc will be in A^* so long as the input vector S remains within the region $\{(s_1, s_2) | 4 \leq s_1 \leq 7, 8 \leq s_2 \leq 11\}$. A similar region exists for each of the elements in A^* . The selection of each element in A^* corresponds to the instantiation of the input vector within a region of input space. The extent of these regions is dependent on the quantization resolution of the intermediate variables m_i .

Any two input vectors which lie close together in input space will occupy many of the same regions. This gives CMAC its property of generalization, i.e. the tendency to produce similar outputs for similar inputs. If $S_1 \rightarrow A_1^*$ and $S_2 \rightarrow A_2^*$, then the amount of generalization between S_1 and S_2 corresponds to the number of elements in the intersection $A_1^* \cap A_2^*$. For example, in Fig. 12, S_2 selects three out of four association cells that S_1 does. Thus the output $h(S_2)$ will differ from $h(S_1)$ only by the contents of the single weight which is not in the intersection $A_1^* \cap A_2^*$.

The nature of the $S \rightarrow M$ mapping causes the number of elements in $A_1^* \cap A_2^*$ to decrease as the input-space distance between the two input vectors grows larger. Finally, at some distance the intersection is null and the subsets A_1^* and A_2^* are disjoint. At that point S_2 can be said to be outside the neighborhood of generalization of S_1 . The value of the function $h(S_2)$ is then independent of $h(S_1)$.

4.3. THE $A \rightarrow p$ MAPPING

In the cerebellum, axons from the granule cells contact a large number of Purkinje cells directly through a set of weighted synaptic connections and indirectly through inverting interneurons. In CMAC all of the Purkinje cells with the same $S \rightarrow A$ mapping are lumped together as a single element of the output vector. Thus, each element of the output $P = (p_1, p_2, \dots, p_L)$ is computed by a separate CMAC from the formula

$$p_k = \sum_i a_i^k w_i^k,$$

where $A^k = (a_1^k, \dots, a_m^k)$ is the association cell vector of the k th CMAC and

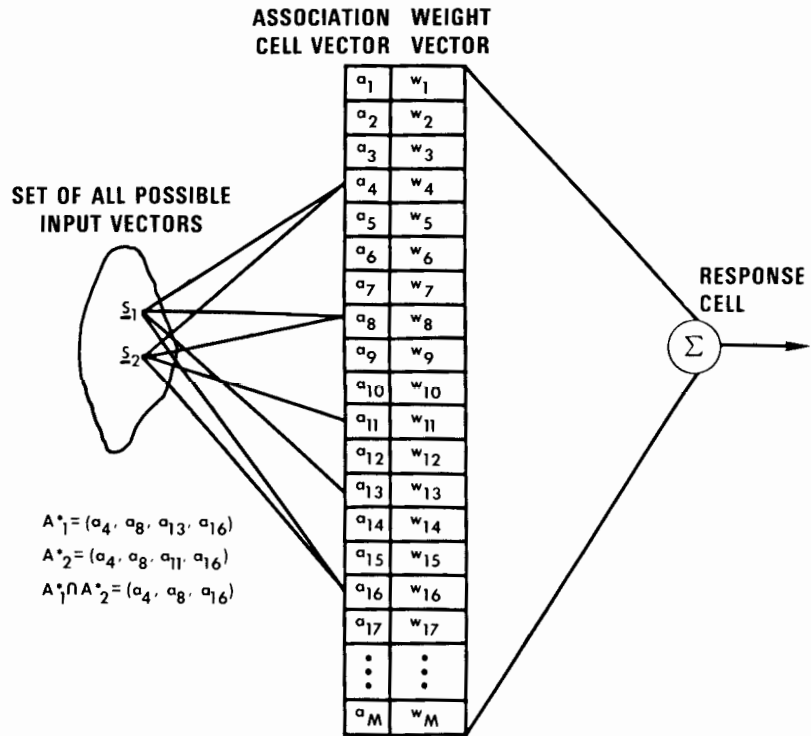


FIG. 12. CMAC's ability to generalize derives from the overlap, or intersection, $A_1^* \cap A_2^*$. The response cell is affected in the same way for both input patterns S_1 and S_2 by all weights in the intersection $A_1^* \cap A_2^*$. Only if the intersection is null will the outputs be independent.

$\mathbf{W}^k = (w_1^k, w_2^k, \dots, w_m^k)$ is the weight vector of the k th CMAC. Of course, only the nonzero elements of A^k which constitute A^{k*} contribute to the output p_k .

In the example in Fig. 11 there is only one output p , and the weights pointed to by A^* are

$$w_{Bc} = 1.0,$$

$$w_{Hj} = 2.0,$$

$$w_{Pq} = 1.0,$$

$$w_{Vc} = 0.$$

These weights are summed to produce the output

$$p = 4.0.$$

Thus, the input $S = (7, 10)$ produces the output $h(S) = 4$. The entire set of weights shown in Fig. 11 defines the function in Fig. 13.

At every point in input space, four weights are selected whose sum is the value of the output. As the input vector moves from one point in input space to an adjacent point, one weight drops out to be replaced by another. The new weight minus the old is the difference in the values of the output at the two adjacent points. Thus, the difference in adjacent weights is the partial derivative (really the partial difference) of the function at that point.

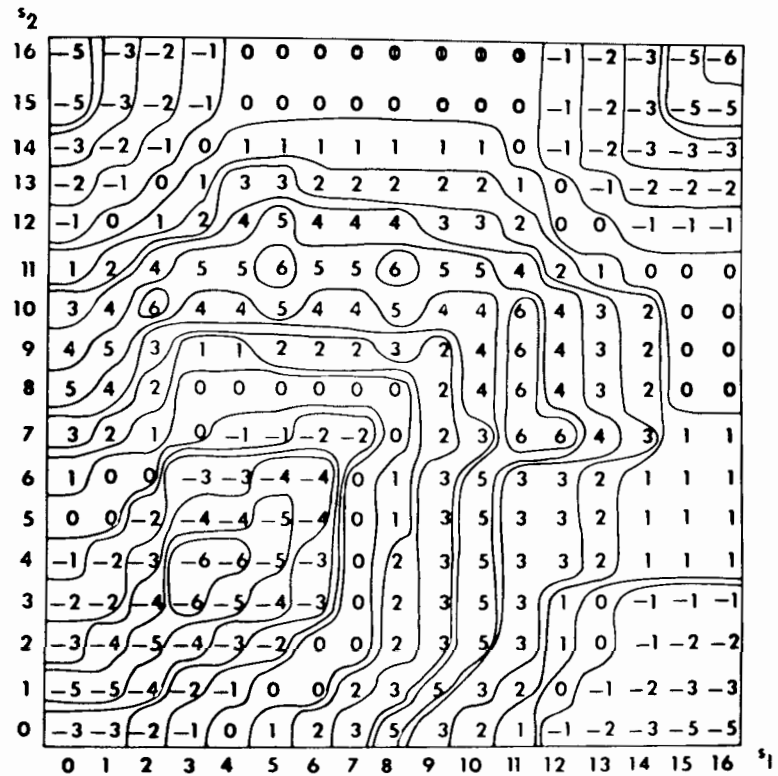


FIG. 13. A two-input function of the type which CMAC can readily store. The particular set of weights in Fig. 11 produces the above function.

For example, in Fig. 11, if the input vector moves from $S=(7,10)$ to $S=(8,10)$ the weight $Bc=1.0$ drops out and is replaced by $Cc=2.0$. The value of the output thus changes from 4 to 5.

4.4. DETERMINATION OF THE H FUNCTION

In the cerebellum each Purkinje cell has a unique fiber, called a climbing fiber, which is believed to be related to learning. There have also recently been discovered fibers from the locus ceruleus which may also be involved in learning [20]. While the exact mechanisms for memory storage are as yet unknown, it has been hypothesized—and some recent evidence suggests [35]—that climbing fibers carry error-correction information which punishes synapses which participate in erroneous firings of the Purkinje cell.

A procedure for entering a function in CMAC is as follows:

(1) Assume that \hat{H} is the function we want CMAC to compute. Then $\hat{P}=\hat{H}(S)$ is the desired value of the output vector for each point in the input space.

(2) Select a point S in input space where \hat{P} is to be stored. Compute the current value of the function at that point $P=H(S)$.

(3) For every element in $P=(p_1, p_2, \dots, p_L)$ and in $\hat{P}=(\hat{p}_1, \hat{p}_2, \dots, \hat{p}_L)$, if

$$|\hat{p}_i - p_i| \leq \xi_i,$$

where ξ_i is an acceptable error, then do nothing; the desired value is already stored. However, if

$$|\hat{p}_i - p_i| > \xi_i,$$

then add to every weight which contributed to p_i the quantity

$$\Delta_i = \frac{\hat{p}_i - p_i}{|A^*|}, \quad (1)$$

where $|A^*|$ = the number of weights which contributed to p_i .

At present there exists no formal criterion for the convergence of this procedure. Convergence clearly depends on the size of the acceptable error band ξ_i . It also depends on the relationship between the size of the neighborhood of generalization and the highest spatial frequency component in the desired function \hat{P} .

An example of how an arbitrary function such as $\hat{p} = \sin x \sin y$, where

$$x = 2\pi s_1 / 360,$$

$$y = 2\pi s_2 / 360,$$

can be stored in CMAC is shown in Figs. 14–16. In this example the input is defined with unity resolution over the space

$$S = \{(s_1, s_2) | 0 < s_1 \leq 360, 0 < s_2 \leq 180\},$$

and the number of weights selected by each input is

$$|A^*| = 32.$$

In this case CMAC will generalize so that any two input vectors which differ by only one resolution element will have 31 weights in common. Not until two input vectors are 32 resolution elements apart will they map into a pair of disjoint subsets A^* .

Initially, the weights were all set to zero and the point $S_1 = (90, 90)$ was chosen for the first data entry. The value of the desired function at $(90, 90)$ is 1. By Eq. (1) each of the weights selected by $S_1 = (90, 90)$ is set to $\frac{1}{32}$, causing the proper value to be stored at S_1 as shown in Fig. 14. Following this operation we find that a second input vector $S_2 = (91, 90)$ will produce the output $\frac{31}{32}$. This is because S_2 shares 31 weights with the vector S_1 . A third vector $S_3 = (92, 90)$ [or $S_4 = (90, 92)$] will have an output $\frac{30}{32}$ because of sharing 30 weights with S_1 , etc. The result is that the CMAC memory generalizes. Adjacent memory locations are not independent, and a plot of values stored at each point in input space has the appearance of a stretched rubber sheet. Pulling one point to a particular value, as in Fig. 14, affects adjacent points.

Generalization has the advantage that training (or data storage) is not required at every point in the input space in order for an approximately correct response to be obtained. This means that a good first approximation to the correct H function can be stored for a sizable envelope around an AND trajectory by training at only a few points along that trajectory. For example, Fig. 15 demonstrates the effect of training at only 16 points along the trajectory

$$T_s = \{(s_1, s_2) | 0 < s_1 \leq 360, s_2 = 90\}.$$

Further training at 175 points scattered over the entire input space $S = \{(s_1, s_2) | 0 < s_1 \leq 360, 0 < s_2 \leq 180\}$ produced the result shown in Figure 16.

Generalization enables CMAC to predict on the basis of a few experiences what the appropriate behavioral response will be in a similar situation. Needless to say, such predictions are not always correct and sometimes need to be corrected by further learning. Nevertheless, generalization

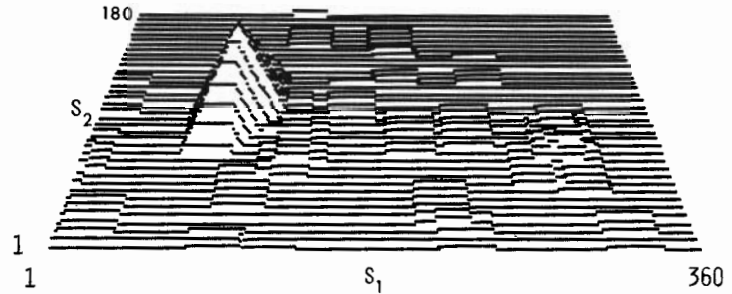


FIG. 14. An illustration of the neighborhood of generalization around a CMAC input vector. In this example the value 1 stored at $S = (90, 90)$ generalizes to neighboring points in input space. The amount of generalization falls off with distance from the stored input. This causes a plot of the CMAC output to appear like a stretched rubber sheet.

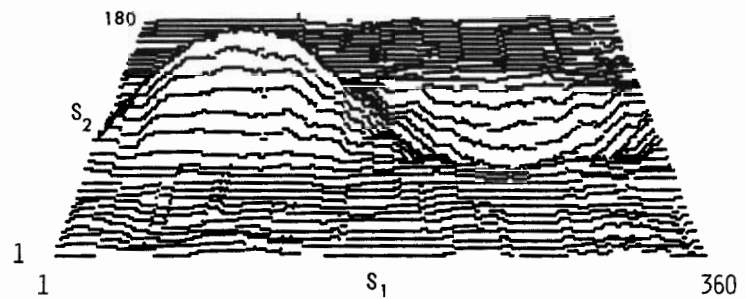


FIG. 15. The effect of training with the function $\hat{p} = \sin x \sin y$ at 16 points along the $s_2 = 90$ axis.

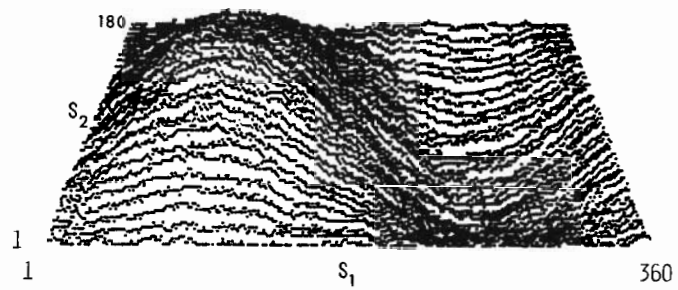


FIG. 16. The effect of training with the function $\hat{p} = \sin x \sin y$ at 175 points over the entire range of the input variables.

is essential in order to cope with real-world environments where the identical set of AND/OR trajectories never recurs.

The ability of the CMAC H function to discriminate (i.e. produce different outputs for different inputs S_1 and S_2) depends upon how many weights in A_1^* are not in A_2^* and how different in value those weights are. If two inputs which are close together in input space are required to produce significantly different outputs, then repeated training may be required to overcome the (in this case erroneous) tendency to generalize by building up large different values in the few weights which are not in common.

The extent of the neighborhood of generalization is a function of the number of elements in the set A^* and the resolution of the $s_i \rightarrow m_i$ mapping. These may be selected by the system designer to model the properties of various regions of the brain.

4.5. MEMORY REQUIREMENTS

Using conventional methods, computation of a function by table lookup procedures requires R^N memory locations for N variables where each variable can take on R distinguishable values. CMAC, however, does not require a unique table entry for each possible input vector. It maps each input into a set of memory locations such that the amount of memory required is substantially reduced. For example in Fig. 11, the input space is a 17×17 array, or 289 potential input vectors. The CMAC memory, however, has only 100 elements.

For larger input spaces such as are encountered in real-world computations, hash coding techniques are representative of the neurological interactions used to compress much further the number of physical memory locations required to store a function. In Figs. 14–16 the 360×180 -element input space was represented in a 1024-location memory by hash coding. An extensive discussion of these hash coding procedures and the neurophysiological processes they represent appears elsewhere [1, 4].

Hash coding "collisions" introduce noise which is distributed randomly over the entire input space. Hashing noise can be seen in the base plane in Figures 14 and 15. Each time new data are stored, previously stored data are degraded somewhat. Thus, the H function is most precisely defined in the regions where training is most recent, and gradually degrades due to learning interference in regions where training is not recent. In this sense CMAC, like the brain, tends to forget.

In the cerebellum of the cat, where most of the experimental data were obtained, each output Purkinje has about 200,000 synaptic weights and its interneurons have about that many again. Furthermore, as many as 100 Purkinje cells may be driven by essentially the same $S \rightarrow A^*$ mapping. Thus, the physical memory for each cerebellar CMAC may be as high as 40 million weights. This is quite an adequate-size memory to store functions of

several tens of input variables. This allows the cerebellum to include in its control computations many parameters with subtle effects such as flexing of structural components, nonlinearities in actuator drive, and Coriolis forces due to moving inertial frames.

The large number of weights used by the cerebellum does not, however, preclude the use of CMAC as a practical device for servo control. A CMAC with 7168 words of memory has been used to control a six-axis robot manipulator [4], and one with 2047 weights has successfully controlled a two-degree-of-freedom biped walking device [9]. Even smaller CMACs such as the one shown in Fig. 11 may have many applications for servo control. A CMAC with up to 12 input variables can readily be implemented with microprocessor technology.

4.6. GENERALITY

The anatomical structure of the cerebellum happens to be particularly suitable for neurophysiological experiments, making it feasible to collect sufficient quantitative data for the construction of mathematically tractable models such as CMAC. There is, however, little reason to believe that the cerebellum is fundamentally different from many other parts of the brain in the manner in which information is processed and stored. The basic structures of a large output cell (sometimes called a principle, relay, or projection neuron) served by a cluster of interneurons is quite typical throughout the brain [53]. These clusters commonly receive input from a large number of nonspecific neural fibers similar to the mossy fibers in the cerebellum. In many instances they also receive specific inputs from fiber systems which are more or less analogous to climbing fibers in the cerebellum. It is also known that the basic neural interactions between fiber and cell systems is quite similar throughout the brains of all mammals. For example, in the spinal cord the motor neurons are the principle neurons and the Renshaw cells are the interneurons. In the neocortex the principle neurons are the pyramidal cells and the interneurons are of several different types. Specific and nonspecific input fibers to the neocortex differ in their points of origin and in their mode of interaction with principle neurons and interneurons. Also in the thalamus, as well as in the olfactory bulb, the hippocampus, and even to a large degree in the retina, there is the familiar architectural pattern of a principle neuron, served by a group of interneurons with a more or less distinct separation between specific and nonspecific input fibers [53].

There are also, of course, many differences from one region of the brain to another in the character of these cell systems and in their interactions. In addition there are complicating factors such as axon-axon and dendrite-dendrite synaptic connections, which play undoubtedly important but unknown roles in information processing, storage, and retrieval in the brain.

Nevertheless, there are clear regularities in organization and similarities in function from one region to another which are significant and which suggest that, at least to a first approximation, the basic processes are similar. The implication is that the basic model of information processing suggested by CMAC (i.e. the concept of a group of principle neurons together with their interneurons transforming an input vector into an output vector in accordance with a mathematically definable relationship) may be useful in analyzing the properties of many different cortical regions and subcortical nuclei.

5. COMPUTATIONAL CAPABILITIES

The ability of CMAC to store and recall a very general class of multivariant mathematical functions is an extremely powerful result. It suggests a single elementary mechanism by which the central nervous system can perform many different complex operations.

5.1. ARITHMETIC COMPUTATION

The ability of CMAC to compute $P = H(S)$ demonstrates how a relatively small cluster of neurons can calculate the type of mathematical functions required for multivariant servos, coordinate transformations, and task decomposition operators. These are minimum requirements for the activities of running, jumping, throwing, catching, and flying which are accomplished with apparent ease by the tiniest rodents, birds, and even insects. In the simplest of creatures, the weighted synaptic connections which produce the H function may be genetically predetermined with only minor adjustments left to be accomplished through a trial-and-error learning process. At the lower levels of the motor hierarchy these predetermined circuits correspond to reflexes. At higher levels they produce the types of behavior known as instinct.

5.2. CONDITIONAL BRANCHING

The multivariant nature of CMAC means that one or more of the input variables in S can be used to select different regions in input space where entirely different functions may be stored. Such variables can thus produce the equivalent of a conditional branch in a computer program.

Assume for example that in Fig. 16 a third variable s_3 had been included in the input vector. Assume that s_3 is held constant at $s_3 = 0$ while storing the function $p = \sin x \sin y$. Following that, an entirely different function, say $p = x + 9y^2$, could be stored with s_3 held constant at $s_3 = 50$. Since every point in the input space for $s_3 = 0$ is outside the neighborhood of generalization of the input space for $s_3 = 50$, there would be no interference except for

random hash-coding collisions. The stored function would then be

$$\begin{aligned} p &= \sin x \sin y & \text{if } s_3 = 0, \\ p &= x + 9y^2 & \text{if } s_3 = 50. \end{aligned}$$

In the interval $0 < s_3 < 50$ the function would change smoothly from $p = \sin x \sin y$ to $p = x + 9y^2$. Additional functions could be stored for other values of s_3 . Other conditional variables s_4, s_5, \dots might also be used to select additional functions.

If these conditional variables are part of the command vector, then each different input command can select a different stored function. If they are part of the feedback, then different environmental conditions can trigger entirely different response patterns.

5.3. FUZZY-STATE AUTOMATA

In the cerebellum, as in many other parts of the brain, fibers from output cells send branches directly back into their own interneuron clusters to become inputs. For CMAC this implies that some of the elements in the output vector \mathbf{P} may loop back to become elements in the input vector \mathbf{S} as shown in Fig. 17.

An H function can now be defined such that for binary inputs the CMAC behaves like a finite-state automaton. In this case the feedback inputs s_1 and s_2 define the state of the machine, and s_3 corresponds to the input.

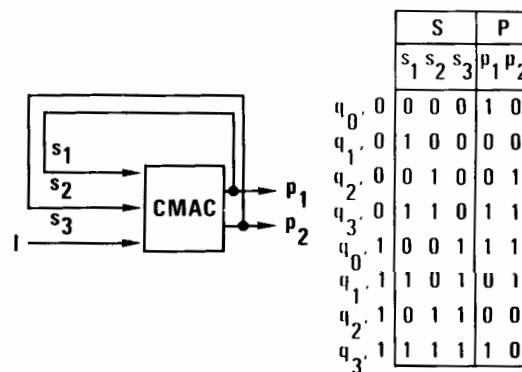


FIG. 17. Feedback loops can transform CMAC into the equivalent of a finite-state automaton. The feedback inputs s_1 and s_2 define the state of the machine, and s_3 corresponds to the input. The transition table is determined by the CMAC weights.

If the $S \rightarrow A$ mapping for the machine in Fig. 17 is chosen similar to that in Fig. 11 (except for the range of the variables, which extends from 0 to 1 rather from 0 to 16), then Fig. 18 shows the input space and the value of the output P for points along lines connecting the vertices representing the binary input vectors.

The CMAC in Fig. 17 is equivalent to the finite-state automaton diagrammed in Fig. 19. In general it is possible to construct the CMAC equivalent of any finite state automaton. Of course, the CMAC can accept inputs and produce outputs which are not binary values. CMAC is thus a sort of fuzzy-state automaton. CMAC reduces to a regular finite automaton in the limiting case where the resolution on the s_i inputs goes to one (i.e., where each set $\{m_i^*\}$ contains only one binary variable).

Assume in Fig. 17 that the input variable s_3 is the command vector C , and the feedback variables s_1 and s_2 are the vector F . It is now clear how a neural cluster modeled by CMAC can generate a string of outputs in response to a constant (unchanging string of) input command(s). A different unchanging string of input commands generates a different output

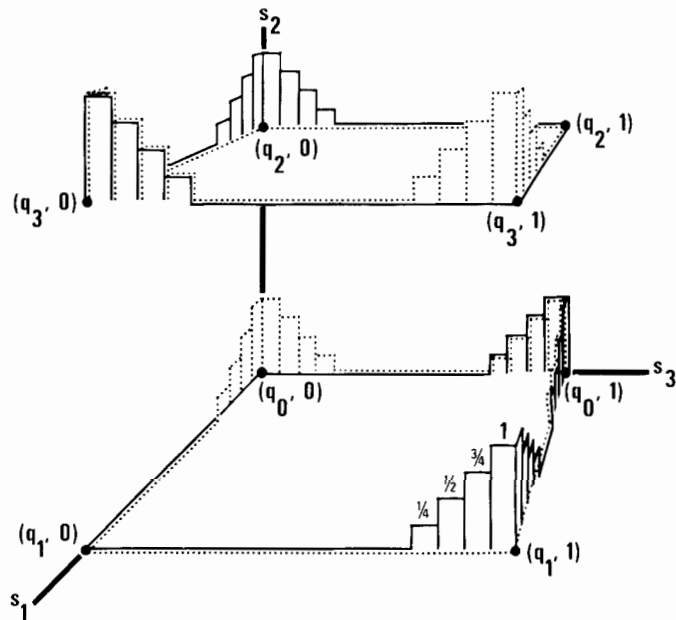


FIG. 18. The input space of the CMAC in Fig. 17 with the value of the output P for points along the lines connecting the vertices. The dotted lines correspond to p_1 and the solid to p_2 .

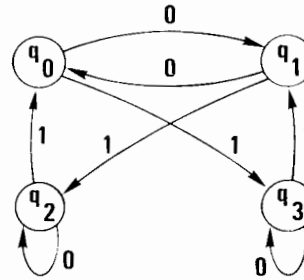


FIG. 19. The state diagram for a finite-state automaton for which the CMAC in Figure 17 is the equivalent.

string or trajectory. Additional variables added to F from an external source, or context, increases the dimensionality of the input space and thus can alter the output string in a "context-sensitive" way.

This configuration of the CMAC model is relevant to a long-standing controversy in neurophysiology regarding whether behavior patterns are generated by "stimulus-response chaining" (i.e. a sequence of actions in which feedback from the sensory organs is required to step from one action to the next) or by "central-patterning" (i.e. a sequence of actions which is generated by internal means alone) [14]. A CMAC hierarchy may have tight feedback loops from the output of one level back to its own input, as well as longer internal loops from one level to another in the same hierarchy. It may in addition have feedback from the environment to alter the centrally patterned movements in accordance with environmental conditions.

The capability of CMAC to simulate a finite-state automaton, to execute the equivalent of conditional branches, and to compute a broad class of multivariant mathematical functions makes it possible to construct the CMAC equivalent of a simple computer program. Alternatively, it is possible to construct a program equivalent of each H module in a CMAC hierarchy. A hierarchically structured robot control system based on CMAC theory is currently under development by Anthony Barbera at the National Bureau of Standards [7].

5.4. PATTERN RECOGNITION

The computation of the CMAC output by the formula

$$p = \sum_i a_i w_i$$

means that p is equivalent to a linear discriminant function on the A vector. If the input vector S is a pattern, then the addition of a threshold device to

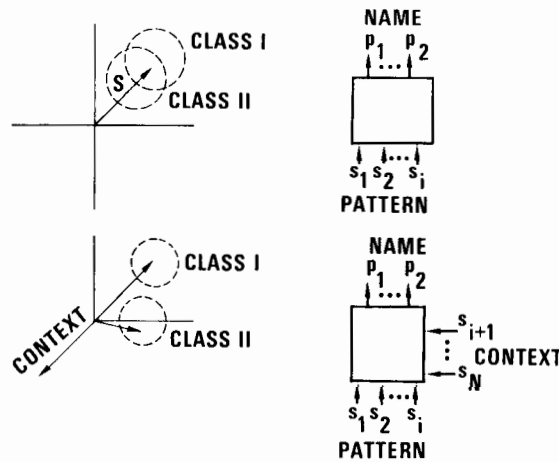


FIG. 20. Patterns in two overlapping or ambiguous classes may be inseparable without the addition of context variables. Context can increase the dimensionality of the input space and disambiguate overlapping classes.

the output is all that is required to turn CMAC into a classical pattern recognizer.

The $S \rightarrow A$ mapping gives CMAC the ability to restrict the neighborhood of generalization on the input space. This means that the A^* for any two regions in input space can be made disjoint as long as the regions are nonoverlapping. Thus, CMAC can be trained to perform a very broad class of pattern-recognition tasks.

If we assume that a CMAC pattern classifier includes in its input S a set of context variables as well as a set of pattern variables, as shown in Fig. 20, then the context vector can be used to shift the total input vector to different parts of input space depending on the context. Thus an ambiguous pattern can be recognized as in class I when accompanied by context I, and in class II when accompanied by context II.

5.5. ASSOCIATIVE MEMORY

Assume CMAC is embedded in a sensory data pathway such that the training input \hat{P} to the CMAC carries sensory data, as shown in Fig. 21. The \hat{P} input alters the synaptic weights of the active parallel fibers so that the difference between the training input \hat{P} and the CMAC memory P is reduced or nulled. The effect is that the sensory data carried on the training input as \hat{P} are now stored in an address which is the S vector present when the training data occurred. Any time this same S vector recurs in the future, the CMAC will output the stored sensory data which were previously

present on the sensory pathway. The precision with which the sensory data can be recalled is proportional to the precision with which the S address is reproduced. The sensory data are thus stored "in association with" the S address which was present when they occurred. In Fig. 21 the S addresses are derived from the behavior-generating hierarchy. Thus, sensory data are stored in association with the P vectors in the generating hierarchy that were present while the data was being stored. If these S addresses trace out a behavior trajectory T_s while the sensory data are changing with time, then an entire sensory experience will be stored as a sequence of recalled P vectors in association with the sequence of S -vector addresses which constitute the behavior trajectory T_s . If at a later time the S vector traces out the same trajectory T_s (as, for example, when the same action sequence is being generated by the generating hierarchy), then the sensory experience will be replayed much like a tape recording. Again the fidelity of recall will depend on the accuracy with which the S vector retraces the storage trajectory T_s .

If, during learning, the gain factor in the weight adjustment procedure is small, many passes along T_s may be required to record a sensory experience with a good signal-to-noise ratio.

In such a memory system, the alteration of synaptic weights is thought to be accomplished by a series of chemical reactions caused by synaptic

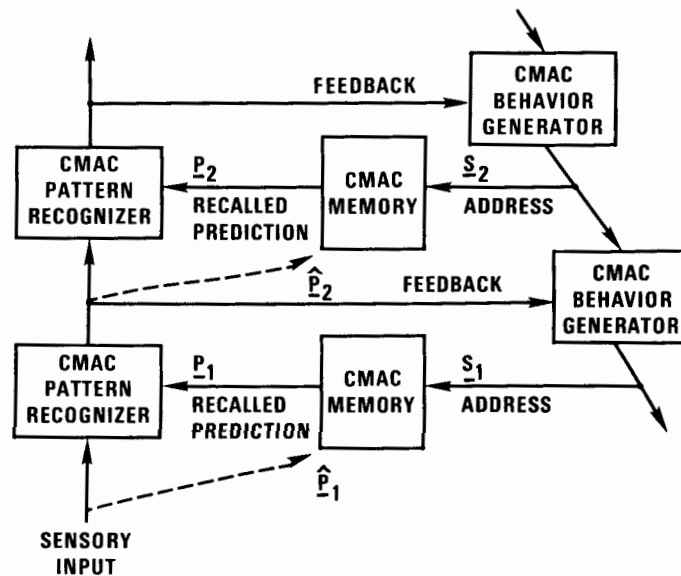


FIG. 21. Sensory experiences may be stored "in association with" motor behavior if the S inputs to memory CMACs are derived from the generating hierarchy and the \hat{P} "desired outputs" are derived from the sensory-processing hierarchy.

transmitter substances released by the active fibers. The decay phenomenon of short-term memory may simply be the exponential decay of concentration of these transmitters. The consolidation of long-term memory appears to involve protein synthesis causing permanent growth at the synaptic sites. A more extensive discussion of this type of associative memory is contained in [5].

6. HIERARCHIES IN THE BRAIN

There is considerable neurophysiological and anatomical evidence that hierarchies of CMAC-like processing modules similar to that shown in Fig. 5 actually exist in the neurological substrate of the central nervous system, although certainly not in such a simple schematic form.

6.1. THE DESCENDING MOTOR HIERARCHY

At the very bottom of the motor hierarchy are the motor neurons which Sherrington called the "final common path." These motor neurons actually drive the muscles, and their output firing rate can be considered the terminal trajectory of the AND/OR tree. The motor neurons and their associated interneurons receive feedback directly from stretch receptors in the muscles being controlled. This lowest-level feedback loop, the stretch reflex or "gamma loop," is the most studied and best understood of all the levels in the hierarchy.

A CMAC representing a spinal motor neuron and its interneurons receive feedback F from stretch receptors via the dorsal roots as well as from other motor neurons reporting ongoing activity in related muscles. The command vector C to this lowest level comes from the vestibular system, which provides inertial reference signals necessary for balance and stability [69], as well as from the reticular formation and basal ganglia [22], and in primates also directly from the motor cortex [8]. Much of the vestibular system input passes through, or is modulated by, the cerebellum, which receives feedback from joint position sensors, tendon tension sensors, and skin touch sensors [42]. Thus, parts of the motor cortex, particularly sources of the pyramidal fibers, together with the cerebellum and basal ganglia represent a second level in the motor hierarchy. The structure of the command and feedback inputs to the CMACs representing the cerebellum has already been extensively discussed.

The motor cortex contribution to the second level has been called the transcortical servo loop by Phillips [44]. Evarts and Tanji [13] have observed cells in the motor cortex whose response P to a stretch stimulus F can be altered (indeed, completely reversed) by different command inputs C . An experimental animal was trained to pull a lever upon feeling a jerk if a red light preceded the stimulus, and push the lever if a green light preceded the stimulus. Both the command C (high firing rate = red, low = green) and the

altered response **P** (pull if **C**=high, push if **C**=low) were observed. There is a measurable time delay which clearly separates the effect of feedback to the lowest level (10–20 msec), feedback to the second level (30–50 msec), and changes in command inputs to the second level (100–200 msec) [13, 54].

Other experiments by Evarts [15, 16] and Thach [60] have shown that neurons in the cerebellum, thalamus, and motor cortex alter their firing rates at various intervals prior to learned movements, and well in advance of any response feedback. This suggests the propagation of goals and subgoals down the motor hierarchy as the various levels receive commands and issue subcommands in preparation for the initiation of a task.

Further evidence that hierarchical structures exist and function as AND/OR task decomposition operators in the generation and control of motor behavior can be found in almost any neurophysiological textbook. For example, brain-stem transection experiments have demonstrated that stereotyped elemental movements such as extension, rotation, and turning movements of the head and body are generated and controlled in the mesencephalic and lower diencephalic brain stem [22]. Rotational movements of the head and eyes are controlled by the interstitial nucleus, raising movements of the head and body by the prestitial nucleus, and flexing movements by the nucleus precommissuralis. Turning movements of the entire body, which are coordinated strings of these elemental movements, do not occur unless the higher brain-stem regions of the pontile and mesencephalic reticular formation are intact. Fully expressed coordinated circling movements require the candate nucleus and the cingulate gyrus of the cerebral cortex [18].

The candate nucleus is part of the basal ganglia. Disease of another part of the basal ganglia, the striate body, can cause a person to perform a perfectly normal pattern of movements for a few seconds and then suddenly switch to a different pattern, and then to another. This disease is commonly known as St. Vitus' dance [11].

The precise identification of higher levels in the motor hierarchy becomes increasingly difficult because of the enormous increase in complexity of behavior which results from each additional level of control. Furthermore, it is clear that the hierarchies which actually exist in the nervous system are by no means schematically simple. The motor system is certainly not a single linear hierarchy, but a multiplicity of partially interrelated partly redundant hierarchies. The pyramidal system which links the motor cortex directly to the lowest level motor neuron circuitry exists in parallel with the extrapyramidal system which involves the red nucleus, the cerebellum, and the vestibular nuclei [42]. These both interact with the postural control system of the basal ganglia and reticular formation. There are strong indications that at least four different hierarchies exist in the language area, one for each of reading, writing, speaking, and hearing [57].

There are clearly separate hierarchies for the right and left sides of the brain with cross-links through the corpus callosum. Furthermore there are many looping structures, which may or may not fit neatly within the framework of a linear hierarchy. The motor cortex transmits to the pons, which transmits to the cerebellum. The cerebellum transmits to the thalamus and then back to the motor cortex. These regions are by no means homogeneous, and there is no reason to assume that all cells in the same anatomical structure occupy the same functional level of a hierarchy. Such loops may simply connect various levels in the hierarchy which are microscopically, and hence logically, distinct even though they occupy the same macroscopic neural structures.

6.2. THE ASCENDING SENSORY HIERARCHY

Different types of feedback inputs at each level of the descending motor hierarchy imply the existence of an ascending hierarchy of information-processing operators. At each level of the ascending hierarchy, input vectors and trajectories correspond to sensory data which are transformed by pattern-recognition operators into output vectors and trajectories corresponding to recognized features. Here, the existence of an output vector within a particular region of output space corresponds to the recognition of a particular event.

There is considerable independent neurophysiological evidence for ascending sensory processing hierarchies. Hubel and Wiesel [24] have demonstrated increasingly sophisticated pattern recognizers in at least four clearly distinguishable hierarchical levels in the visual system. Similar sensory processing hierarchies have been studied in the auditory system [19] and also in the proprioceptive and kinesthetic pathways [42]. Use of this information for controlling behavior clearly requires cross-couplings from the sensory processing hierarchy to the motor generating hierarchy. At each level output vectors form inputs to the next higher processing stage as well as feedback to the parallel generating hierarchy.

Signal-detection theory predicts that the efficiency of the sensory processing hierarchy should be enhanced if there are complementary links from the motor generating hierarchy to the sensory processing hierarchy. This type of information pathway is what researchers such as von Helmholtz [62] and later Sperry, [58] von Holst and Mittelstaedt, [63] and others [14] have called an "efference copy." This information gives the sensory processing hierarchy *a priori* knowledge of what the motor generating hierarchy is doing so that, among other things, sensory signals resulting from movement of the body can be distinguished from sensory signals resulting from movement of objects in the environment. This, for example, enables the visual system to distinguish between a rotation of the eyes and a rotation of the room about the eyes. It shifts the output trajectories of

processing CMACs into different recognition regions depending on what the motor generating hierarchy is doing while sensory data are being processed.

If the links from the generating hierarchy to the processing hierarchy include associative memory modules as shown in Fig. 21, then these pathways also provide the sensory system with a memory trace of what

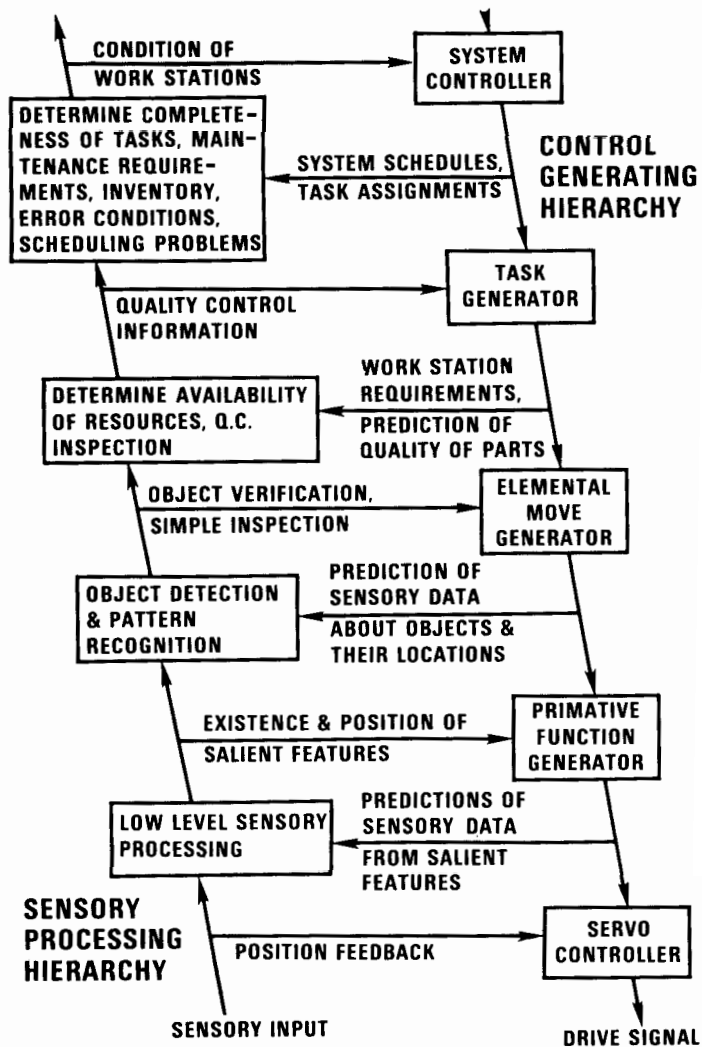


FIG. 22. A processing-generating hierarchy developed for the control of advanced sensory-interactive industrial robotic systems.

sensory data had occurred on previous occasions when the motor generating hierarchy was in the same or a similar state along a similar trajectory. This provides the sensory processing system with predictions of what sensory data to expect. This in turn enables the sensory hierarchy to do predictive filtering, to recognize unexpected events (as well as the absence of expected events), and to perform context-sensitive recognitions.

Figure 22 shows a cross-coupled processing-generating hierarchy which is being developed at the National Bureau of Standards for robot control. It illustrates the kinds of information that flow back and forth between the processing-generating hierarchies in an advanced sensory-interactive industrial robot system.

6.3. LOOPS AND RHYTHMS

Cross links between the generating and processing hierarchies produce a series of loops. Analysis of information flow in such a looping structure resembles the analysis of phase-lock loops. This is illustrated in Fig. 23. Inputs from various levels of the generating hierarchy behave in many ways like local oscillators, or signal predictors. Sensory input is compared against the predicted signal, and systematic errors tend to "pull" the generating hierarchy into synchrony with trajectories in the sensory stimuli. Von Holst called this the "magnet effect" [64]. When synchrony is achieved, the processing-generating hierarchy "locks on" to the sensory input. This lock-on phenomenon corresponds to detection or recognition of temporal patterns in the incoming signals.

Assume for example that the two CMACs in Fig. 23 are a phase-lock loop such that the input **PATTERN** is a signal $f(t)$ and the **PREDICTION** is another signal $f(t - \tau)$. If the processing CMAC on the left computes the product **PATTERN** \times **PREDICTION**, then the output **NAME** is $f(t)f(t - \tau)$. When τ corresponds to

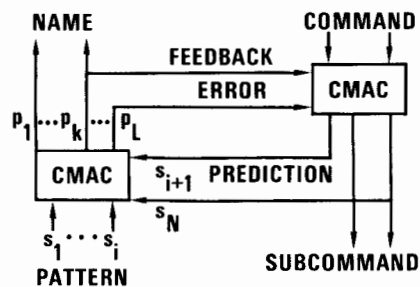


FIG. 23. A pair of CMACs may act as a phase-lock loop. Context can provide information on which to base a prediction. An error signal can refine, or pull, the prediction so as to lock on to the incoming pattern.

$\frac{1}{4}$ of the period of the input $f(t)$, a low-pass filter applied to the output will produce a phase error signal which when applied to the generating CMAC on the right will cause the PREDICTION signal $f(t - \tau)$ to track and lock on to the input PATTERN $f(t)$. A multiplicity of such loops with different delays ($\tau > 0$) or different predictive intervals ($\tau \leq 0$) will produce a multiplicity of outputs which when processed through low-pass filters produce an approximation to an autocorrelation function:

$$R(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f(t)f(t-\tau) dt,$$

such that

$$\begin{aligned} p_1 &= R(\tau_1), \\ p_2 &= R(\tau_2), \\ &\vdots \\ p_{L/2} &= R(\tau_{L/2}), \\ &\vdots \\ p_L &= R(\tau_L), \end{aligned}$$

where $\tau_1 > \tau_2 > \dots > \tau_{L/2} > \dots > \tau_L$ and $\tau_{L/2} = 0$. Wightman [68] has shown that such an autocorrelation function yields a subjective perception of pitch which is in good agreement with psychophysical data. In terms of the diagram in Fig. 23 the existence of an output on element p_i would correspond to the perception of pitch at a frequency $1/\tau_i$. The existence of an output on element $p_{L/2}$ corresponds to a confidence factor indicating the strength of the lock-on.

Figure 24 suggests how a hierarchy of phase-lock loops might interact to generate and recognize language and/or music. In the language model suggested here the intent to communicate a message or to encode a high-level behavioral trajectory into language symbols corresponds to the selection of a top-level input command, or goal. This goal is then decomposed through the H operators at the remaining levels until at the bottom an output string of muscle commands drive the lungs, larynx, lips, and tongue. These H operators embody the syntax, characteristic expressions, and language habits of the speaker. The resulting behavioral patterns, which may involve bodily and facial gestures as well as vocalizations, are thus a product of all the learned and prewired decomposition functions stored throughout the entire processing-generating hierarchy.

In this model, the message to be communicated is encoded by the H functions which generate the language behavior. The understanding of a message received is accomplished when the incoming sensory data can pull

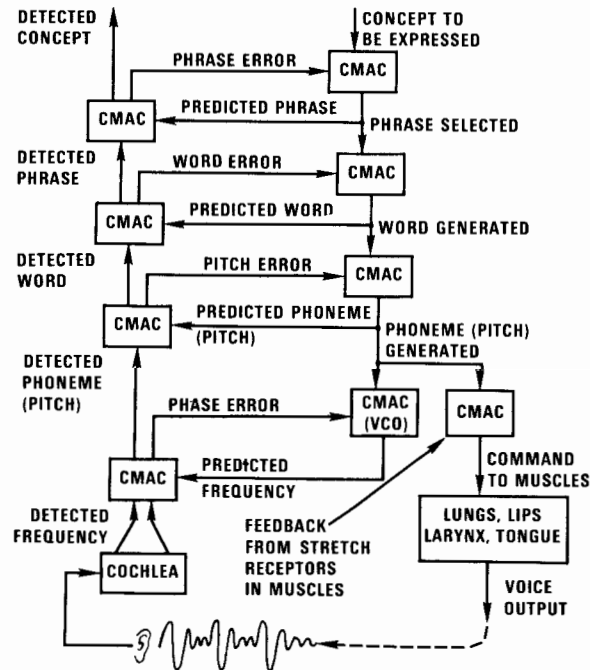


FIG. 24. Cross-coupling at many different levels between motor-generating and sensory-processing hierarchies produces the capability to detect, recognize, and track many different, but related, phrase structures such as exist in speech and music.

the processing-generating hierarchy of the listener into synchrony with the processing-generating hierarchy of the speaker.

In a phase-lock loop, lock-on usually occurs with a positive snap, or "thunk," even if preceded by an extended search. This may correspond to the gestalt experience when we say "Aha!" or "I see!" Once lock-on is achieved the hierarchy can track lengthy sequences of signals even in the presence of noise or interference from similar signals.

Typically vectors at higher levels have a slower time rate of change, and the trajectories defined by these vectors represent greater levels of abstraction. This implies that the higher-level loops lock on to longer-term periodicities. Lock-on at many different levels gives the hierarchy the ability to detect, recognize, and track patterns with many different, but harmonious, phrase structures such as are present in spoken language and music. This may explain the peculiar affinity of the ear for the rhythmic character of poetry and the numerical relationships involved in musical harmony. It may also explain the ability of the ear to ignore bursts of noise and to "flywheel" through auditory dropouts with apparent ease.

The model, with its hierarchy of locked loops, emphasizes the importance of periodic (i.e. repeating) patterns and rhythms. Periodic phenomena are intimately involved in behavior in general and learning in particular for at least four reasons:

(1) Neural activity in the brain, to the extent that it is periodic, provides a predictably recurring pattern of relationships at all levels. This is a prerequisite to recognition, to prediction, to the testing of predictions against observations, and to the learning of systematic variations which form the basis of more sophisticated predictions.

(2) Many learned activities such as walking, running, dancing, singing, speaking, and gesturing have a distinctly rhythmic, and sometimes strictly periodic, character.

(3) Observation of the learning behavior of children reveals that they are particularly fascinated by repetition and periodicities of many different kinds. Ample evidence of this may be found in childrens' songs and games, and in the circumstances accompanying the familiar child's request, "do it again, do it again."

(4) Almost all high-level goal selection (i.e. decisions to engage in common daily activities such as sleeping, eating, attending to personal hygiene, going to school or work, playing, attending social or religious meetings, etc.) is strongly influenced by periodic factors such as sunlight and darkness, work or school schedules, meal times, and each individual's own internal rhythms, which have monthly as well as daily and shorter-term periodicities.

Learning thus takes place against a background of rhythmic patterns which permeate the entire processing-generating hierarchy. Within this framework regularly recurring temporal relationships become recognizable and sequences of events predictable. Recognitions of gradual or small systematic deviations from the predictable lead to the learning of more sophisticated recognitions and predictions. Abrupt departures from the predictable elicit surprise, and large deviations (which are accompanied by intense emotional responses) become memorable events. Continued or prolonged disruption of regular patterns, either in the internal rhythms or in external stimuli, destroy predictability, frustrate learning, and generate emotional stress, which in severe cases can produce neuroses.

6.4. *PERFORMANCE EVALUATION*

Output from the highest-level pattern recognizers interacts with the limbic system, which evaluates the processed sensory data for desirability or undesirability. Olds [40, 41] and many others have located specific regions in the posterior hypothalamus and medial tegmentum which when stimulated produce a pleasurable or rewarding effect. Other regions in the posterior

and lateral diencephalon and lateral tegmentum produce a punishing or aversive effect. These and other emotional centers make a judgement as to the goodness or badness of the state of the world as perceived through the sensory processing system.

Pribram has recently demonstrated that the mediobasal motor cortex is intimately involved both in somatomotor and in viceroautonomic activities [46]. Stimulation as well as lesion experiments in this region strongly suggest that this is an area where entire behavioral sequences (i.e. high-level goals) are selected. The viceroautonomic connection with this region suggests a mechanism by which "gut feelings" influence, and are influenced by, the choice of behavior patterns.

It is also well known that chemoreceptors and hormone detectors and emitters exist in areas such as the hypothalamus and pineal gland, which provide appetitive and sex-drive variables. In addition there are emotional centers of fear, pain, pleasure, sorrow, aggression, and sympathy in the amygdala, the fornix, the hippocampus, and many other regions of the limbic system. These all provide inputs to the goal-selecting levels and training mechanisms of the processing-generating hierarchies [65].

Variables generated by all of these different limbic regions combine with processed sensory data to form the *S* vectors to the highest levels in the motor hierarchy as well as to the autonomic system. These mechanisms can thus decide between one plan and another, one hypothesis and another, or one goal and another. This information can influence the goal selection process directly by providing input variables to the highest-level task decomposition operators, or by providing desired values and reward-punishment data to the training mechanisms which alter the *H* functions.

6.5. PLANNING, FORESIGHT, AND IMAGINATION

In a complex structure of many cross-coupled hierarchies representing many different sensory-motor systems, there are at least three distinctly different modes of operation.

6.5.1. Task Execution Mode. First is the task execution mode, in which the motor-generating hierarchy is committed to a goal and the sensory-processing hierarchy is primarily engaged in providing feedback reporting on the effects of those actions. For example, when speaking, the auditory system is primarily engaged in hearing the sound of the speaking voice and providing feedback for controlling intensity, modulation, and pitch.

6.5.2. Sensory Analysis Mode. A second mode of operation is analysis of sensory data from external sources. When in this mode the generating hierarchy may simply disable its motor output levels or command the lower

levels to execute an overlearned routine task. This leaves the rest of the motor-generating hierarchy available for assisting the sensory-processing hierarchy by creating hypotheses and subhypotheses, as opposed to task and subtasks. Sensory input from the environment is now compared with predictions based on these hypotheses. If the hypotheses are correct, they will be confirmed; if only nearly correct, they can be pulled by feedback from the processing hierarchy. When a high-level hypothesis is successful in generating predictions which match incoming sensory data, the entire processing-generating hierarchy locks on to the incoming sensory data. Sensory input which cannot be correlated with a hypothesis is rejected as "just noise" or "without meaning."

6.5.3. Planning Mode. A third mode of operation is when both the motor output and the sensory input are more or less disconnected from the higher levels of the processing-generating hierarchy. When in this mode the generating hierarchy can set up a sequence of hypotheses and subhypotheses which generate a context, or an associative address. This recalls memories of sensory experiences which were stored at a previous time when a similar context or associative address existed. These internally recalled experiences can then be processed by the rest of the sensory-processing hierarchy as if they were externally derived sensory data.

In this way a person can step through an imaginary task and predict the potential consequences. The generating hierarchy generates what might best be called an idea or a plan. This stimulates the memories of previous sensory experiences associated with that plan, i.e. sensory data stored when a similar task was actually performed. The sensory processing system analyses the recalled data, providing feedback to the generating hierarchy such that it cycles through an entire past experience, or a potential future action. The emotional centers perform an evaluation of these self-induced data. For purposes of planning any number of imaginary tasks can be tried, and the one with the highest evaluation rating (i.e. the most satisfying emotional response) can be selected for actual execution.

This procedure is identical to the familiar artificial-intelligence technique of searching an AND-OR tree for an optimum solution. The heuristics which guide this search are the *H* functions, which are stored throughout the entire cross-coupled processing-generating hierarchy. These constitute a knowledge frame or belief structure which is acquired over a lifetime of experience and education.

Such an internally driven planning-evaluating structure has obvious survival benefits which would tend to drive its evolutionary development. Once developed, however, it may be used for other purposes. For example, the practice of generating imaginary experiences for the sole purpose of producing positive effects on the emotional evaluators constitutes day-dreaming or fantasizing. This third mode of self-induced operation of the

processing-generating hierarchy may also be relevant to the phenomena of dreams and hallucinations.

The complexity and parallelism of the cross-coupled network of processing-generating hierarchies in the human brain makes it possible for more than one mode to be operative in different areas of the brain simultaneously. Nevertheless, no single operator can execute more than one trajectory at a time, and no single hierarchy can be committed to more than one goal at a time.

6.6. LEARNING

The highest-level decision mechanisms and performance evaluators must exist from the very first. Even the most primitive nervous system requires a goal-selecting mechanism from the beginning of its independent existence to commit the organism to food-seeking, danger-avoidance, pleasure-seeking, pain-avoidance, etc. Also as a prerequisite to learning there must be mechanisms which evaluate performance and provide error correcting and reinforcement inputs to the levels being trained. Learning also requires information regarding the level of importance, or emotional arousal, associated with each learning experience. Thus, these highest-level goal-selecting mechanisms must develop early and influence actions throughout life.

In the remainder of the hierarchy, however, motor-sensory learning of specific tasks and skills must develop *at the lowest levels first*. Learning has to be well under way at each level in the hierarchy before it can effectively begin at the next higher level. For a creature with many levels in its processing-generating hierarchy this requires a lengthy childhood so that the mid and upper levels can remain plastic until training at the lower levels is well advanced. It is well known that a child's abilities to deal with abstractions do not develop until after visual processing and motor coordination has been accomplished [45]. Reading does not begin until speech-generating and -recognizing abilities have been developed. [30] This suggests, and there is much experimental and observational evidence to support the notion, that speech and language skills develop in coordination with and via essentially the same mechanisms as hand and body gestures and other motor skills *essential to social communication* [56]. At each level learning must be well developed before consistent and clearly defined trajectories emerge to be used as feedback for the next higher level. Primitive movements, which may be simple adaptations of prewired reflexes, are learned first. Then sequences of commands which string together primitive movements are learned next. As each level in the hierarchy develops, strings of lower-level actions become controllable by a single higher-level command. Strings of strings thus become skills and strings of skills become strategies.

Learning in the sensory processing hierarchy also develops at the lowest levels first. Elementary feature extractors which provide feedback to trigger and control primitive movements must either be prewired genetically or learned first. Recognitions of sequences of features must be learned next. As each level in the hierarchy develops, strings of lower-level recognitions become recognizable as single higher-level events. At all levels sensory learning is accomplished against a background of predictions, hypotheses, and contextual information generated in the motor hierarchy. Error signals for training (as well as for control) are derived from discrepancies between prediction and experience. Understanding of sensory input may be said to occur when long and complex stimuli become predictable, when hypotheses correspond to experience, and when synchrony is achieved between external stimuli and internal expectations at many different levels of the processing-generating hierarchy.

Learning at the lower levels may be primarily through trial-and-error discovery of which behavior patterns are effective or rewarding and which are ineffective or punishing. Higher-level learning, however, comes mostly from imitation of others or from education by teachers. Most of the knowledge required for higher-level learning in humans is derived from the social environment. Society passes on successful strategies which have been acquired over centuries and millenia of painful trial and error through its customs and taboos, its literature, its religious teachings, and its scientific beliefs.

7. SUMMARY AND CONCLUSION

A theory has been set forth wherein the control mechanism for generating motor behavior is *the medium of thinking and planning*. Language and vision, which have occupied a central position in artificial-intelligence research, are seen to be much more intimately involved with the motor system than has been previously assumed. According to this view, language understanding, visual perception, and other complex sensory analyses are active processes wherein hypotheses, preconceptions, beliefs, and prejudices are superimposed on the sensory data stream by the motor generating system.

Language output skills are seen as not essentially different from bodily gestures or other motor skills in how they are generated and learned. Speech is learned like any other motor function when the lower-level primitives of vocalization become commandable and when auditory feedback at the appropriate levels of the hierarchy assumes clearly defined recognition trajectories.

Knowledge is embodied as IF/THEN productions in the transfer functions of the entire cross-coupled hierarchy. These transfer functions select the

goals, generate the predictions, and compute the outputs so as to perceive the world and execute behavior consistent with the stored knowledge.

The CMAC formalism provides a mathematical description for a basic neurological module which can learn, generalize, and compute multivariant functions. The theory of AND/OR goal decomposition suggests how such modules can be assembled into a hierarchical structure to produce sensory-interactive goal-directed behavior. A multiplicity of such hierarchies, controlled at the top by a system which can choose between good and bad, gives rise to a structure in which the abilities to reason abstractly, to learn from the past, to imagine the future, to solve problems, and to plan long-range goals are a direct and natural result.

I am deeply indebted to Dr. Anthony J. Barbera for his advice, suggestions, and many hours of discussion of the concepts presented in this paper. I also thank Miss Debbie Ingram for typing numerous drafts of the manuscript.

REFERENCES

- 1 J. S. Albus, A new approach to manipulator control: the cerebellar model articulation controller (CMAC), *J. Dynamic Systems, Measurement and Control*, Sept. 1975, pp. 220-227.
- 2 —, Data storage in the cerebellar model articulation controller (CMAC), *J. Dynamic Systems, Measurement and Control*, Sept. 1975, pp. 228-233.
- 3 —, A theory of cerebellar function, *Math. Biosciences* 10: 25-61 (1971).
- 4 —, Theoretical and experimental aspects of a cerebellar model, Ph.D. Thesis, Univ. of Maryland, 1972.
- 5 —, A model for memory in the brain, *Cybernetica* XIII: 160 (1970); NASA Tech. Note TN D-6456.
- 6 M. A. Arbib, *The Metaphorical Brain*, Wiley-Interscience, New York, 1972.
- 7 A. J. Barbera, An architecture for a robot hierarchical control system, National Bureau of Standards Publication 500-23, A. J. Barbera, J. Albus, and M. L. Fitzgerald, Hierarchical control of robots using microcomputers, in *Proceedings of the 9th International Symposium on Industrial Robots*, Washington, D.C., Mar. 1979.
- 8 J. M. Brookhart, A technique for investigating central control of posture, in *Neurosciences Research Symposium Summaries*, Vol. 6, MIT Press, Cambridge, Mass., 1972, pp. 60-85.
- 9 P. C. Camana, A study of physiologically motivated mathematical models for human postural control, Ph.D. Thesis, Dept. of Electrical Engineering, Ohio State Univ., 1977.
- 10 R. Davis, B. Buchanan, and E. Shortliffe, Production rules as a representation for a knowledge-based consultation program, *Artificial Intelligence* 8: 15-45 (1977).
- 11 D. Denney-Brown, *The Basal Ganglia: Their Relation to Disorders of Movement*, Oxford U. P., New York, 1962.
- 12 J. C. Eccles, M. Ito, and J. Szentagothai, *The Cerebellum as a Neuronal Machine*, Springer, Berlin, 1967.

- 13 E. Evarts and J. Tanji, Gating of motor cortex reflexes by prior instruction, *Brain Res.* 71: 479-494 (1974).
- 14 E. Evarts, Feedback and corollary discharge: a merging of the concepts, in *Neurosciences Research Symposium Summaries*, Vol. 6 (F. O. Schmidt et al. Eds.), MIT Press, Cambridge, Mass., 1972.
- 15 E. Evarts, Activity of ventralis lateralis neurons prior to movement in the monkey, *Physiologist* 13: 191 (1970).
- 16 E. Evarts and W. T. Thach, Motor mechanisms of the CNS: cerebrocerebellar interrelations, *Ann. Rev. Physiol.* 31: 451-498 (1969).
- 17 R. E. Fikes and N. J. Nilsson, STRIPS: a new approach to the applications of theorem proving to problem solving, *Artificial Intelligence* 2: 189-208 (1971).
- 18 J. D. French, The reticular formation, in *Handbook of Physiology*, (H. W. Magoun, Ed.), Williams & Wilkins, Baltimore, 1960, Sec. 1, Vol. 2.
- 19 R. R. Gacek, Neuroanatomy of the auditory system, in *Foundations of Modern Auditory Theory*, Vol. II (J. V. Tobias, Ed.), Academic, New York, 1972.
- 20 P. Gilbert, How the cerebellum could memorize movements, *Nature* 254: 688-689 (24 Apr. 1974).
- 21 P. H. Greene, Strategies for heterarchical control: an essay, Technical Reports Nos. 7, 8, Computer Science Dept., Illinois Institute of Technology, Chicago, 1975.
- 22 A. C. Guyton, Motor functions of the brain stem and basal ganglia, in *Structure and Function of the Nervous System*, Saunders, Philadelphia, 1976, Chapter 10.
- 23 P. E. Hart et al., Artificial intelligence research and applications, Stanford Research Institute Annual Technical Report on Project 1530, ARPA Contract DAHC04-72-C-0008, 1972.
- 24 D. H. Hubel and T. Wiesel, Receptive fields and functional architecture of monkey striate cortex, *J. Physiol. (London)* 195: 215-243 (1968).
- 25 J. Jackson, *Selected Writings of John Hughlings Jackson* (J. Taylor, Ed.), Hodder and Stoughton, London, 1931.
- 26 H. J. Jerison, *Evolution of the Brain and Intelligence*, Academic, New York, 1973.
- 27 E. R. John, A model of consciousness, in *Consciousness and Self Regulation* (G. E. Schwartz and D. Shapiro, Eds.), Plenum, New York, 1967, Vol. I.
- 28 W. Kohler, *The Mentality of Apes*, Harcourt, Brace, New York, 1925.
- 29 L. Landau and E. Lifshitz, *The Classical Theory of Fields*, Addison-Wesley, Reading, Mass., 1951, Chapters 1, 2.
- 30 E. H. Lonneberg, The natural history of language, in *The Genesis of Language* (F. Smith and G. Miller, Eds.), MIT Press, Cambridge, Mass., 1966.
- 31 D. M. Mackay, Cerebral organization and conscious control of action, in *Brain and Conscious Experience* (J. C. Eccles, Ed.), Springer, New York, 1966.
- 32 Paul D. MacLean, *A Triune Concept of the Brain and Behavior*, Univ. of Toronto Press, Toronto, 1973.
- 33 A. Manning, *An Introduction to Animal Behavior*, Addison-Wesley, Reading, Mass., 1972.
- 34 D. Marr, A theory of cerebellar cortex, *J. Physiol. (London)* 202:437-470 (1969).
- 35 F. A. Miles, personal communication.
- 36 A. Newell, Production systems: models of control structures, in *Visual Information Processing* (W. G. Chase, Ed.), Academic, New York, 1973.
- 37 A. Newell and H. Simon, *Human Problem Solving*, Prentice-Hall, New York, 1972.
- 38 N. J. Nilsson, *Problem-Solving Methods in Artificial Intelligence*, McGraw-Hill, New York, 1971.

- 39 N. J. Nilsson, *Learning Machines*, McGraw-Hill, New York, 1965.
- 40 J. Olds, Differential effects of drive and drugs on self stimulation at different brain sites, in *Electrical Stimulation of the Brain*. (D. E. Sheer, Ed.), Univ. of Texas Press, Austin, 1961.
- 41 J. Olds, Approach-avoidance dissociations in rat brain, *Amer. J. Physiol.* 199:965-968 (1960).
- 42 O. Oscarsson, Functional organization of spinocerebellar paths, in *Handbook of Sensory Physiology: Vol. II. Somatosensory System* (A. Iggo, Ed.), Springer, Berlin, 1970, pp. 121-127.
- 43 J. E. Pfeiffer, *The Emergence of Man*, Harper and Row, New York, 1969.
- 44 C. G. Phillips, Cortical localization and sensorimotor processes at the middle level in primates, *Proc. Roy. Soc. Med.* 66:987-1002 (1973).
- 45 J. Piaget and B. Inhelder, *The Child's Conception of Space*, Norton, New York, 1967.
- 46 K. H. Pribram, Self consciousness and intentionality, in *Consciousness and Self Regulation*, Vol. 1, (G. E. Schwartz and D. Shapiro, Eds.), Plenum, New York, 1976.
- 47 M. R. Quillian, Semantic memory, in *Semantic Information Processing*, (M. Minsky, Ed.), MIT Press, Cambridge, Mass., 1968.
- 48 L. Radinsky, Primate brain evolution, *Amer. Scientist* 63:656-663 (1975).
- 49 B. Raphael et al., Research and applications—artificial intelligence, Stanford Research Institute Final Report on Project 8973, ARPA Contract NASW-2164, Dec. 1971.
- 50 J. W. Rigney and D. M. Towne, Computer techniques for analyzing the microstructure of serial-action work in industry, *Human Factors* 11:113-122 (1969).
- 51 F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan, Washington, 1961.
- 52 E. D. Sacerdoti, A structure for plans and behavior, SRI Artificial Intelligence Center Technical Note 73, 1972.
- 53 G. M. Shepherd, *The Synaptic Organization of the Brain*, Oxford U. P., New York, 1974.
- 54 R. A. Schmidt, The schema as a solution to some persistent problems in motor learning theory, in *Motor Control: Issues and Trends* (G. Stelmach, Ed.), Academic, New York, 1976, pp. 41-65.
- 55 J. Slagle, A computer program for solving problems in freshman calculus (SAINT), Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Mass., 1961.
- 56 F. Smith and G. A. Miller (Eds.), *The Genesis of Language: A Psycholinguistic Approach*, MIT Press, 1966.
- 57 R. W. Sperry, Perception in the absence of the neocortical commissures, in *Perception and Its Disorders*, Research Publication of the Association for Research in Nervous and Mental Diseases, Vol. 48, 1970.
- 58 R. W. Sperry, Neural basis of the spontaneous optokinetic response produced by visual inversion, *J. Comp. Physiol. Psychol.* 43:482-489 (1950).
- 59 Hans-Lukas Tenber, Effects of local brain injury on human behavior, in *The Nervous System: Vol. 2. The Chemical Neurosciences* (D. B. Tower, Ed.), Raven Press, New York, 1975.
- 60 W. T. Thach, Discharge of cerebellar neurons related to two maintained positions and two prompt movements. II. Purkinje cell output and input, *J. Neurophysiol.* 33:537-547 (1970).
- 61 N. Tinbergen, *The Study of Instinct*, Clarendon, Oxford, 1951.

- 62 H. von Helmholtz, *Treatise on Physiological Optics* (P. C. Southhall, Ed. and Transl.), 3rd ed., Vol. 3, Opt. Soc. Amer., Mensha, Wis., 1925.
- 63 E. von Holst and H. Mittelstaedt, Das Reafferenzprinzip, *Naturwissenschaften* 20:464-476 (1950).
- 64 E. von Holst, Entwurf eines Systems der lokomotorischen Periodenbildungen bei Fischen. Ein kritischer Beitrag zum Gestaltproblem, *Z. Vergl. Physiol.* 26:481-529 (1941).
- 65 G. O. Watts, *Dynamic Neuroscience*, Harper & Row, New York, 1975.
- 66 S. L. Washburn, Tools and Human Evolution, *Scientific American* 203:62-75 (Sept. 1960).
- 67 S. L. Washburn and R. Moore, *Ape Into Man*, Little, Brown, Boston, 1974.
- 68 F. L. Wightman, The pattern transformation model of pitch, *J. Acoustical Soc. Amer.* 54:407-416 (1973).
- 69 V. J. Wilson and M. Yoshida, Comparison of effects of stimulation of Deiters' nucleus and medial longitudinal fasciculus on neck, forelimb, and hind limb motoneurons, *J. Neurophysiol* 32:743-758 (1969).
- 70 P. H. Winston, *Artificial Intelligence*, Addison-Wesley, Reading, Mass. 1977.