Chapter 10. Learning

Chapter 10. Learning

10.1 Intelligent Systems and Learning

This Chapter will outline the situation in the area of Intelligent Systems with Learning Control Systems, and will discuss the ideas governing the research in this area. The number of ideas and concepts of learning is very large, neither of them pretends to encompass the whole problem, many of them contradict each other. If one collects them into one document, they do not constitute a consistent body, and are confusing for engineer and researcher working in the field. Areas of adaptive control systems, pattern recognition, production systems, learning controllers and neural networks state different goals of learning and promulgate different approaches to analysis and design of learning systems. Engineers and psychologists, cognitive scientists and logicians-philosophers, computer analysts and mathematicians interpret the term "learning" in a variety of ways slightly different within each discipline.

This situation is not surprising since there exist plenty of different frameworks within different disciplines of science entailed by the different avenues of natural science development. This situation does not seem to be productive for the area of Intelligent Systems. It can be considered detrimental for both the results of scientific research and for the emerging technologies exploring the fruitful area of intelligent control, intelligent robotics, intelligent communications, and others where the idea of making systems learning (in a clear layman understanding of this word) is very promising.

The theme of Learning permeates this book. We discussed already learning in the previous Chapters. In Chapter 2, the primary learning procedures of clustering and classification were introduced, learning semiosis was presented, and the triplet of GFACS was described. In Chapters 3 through 8, elements of learning were demonstrated to be ingrained in the algorithms of knowledge organization, sensory processing, world modeling, and behavior generation. This Chapter is an effort to determine a unified framework for putting together the pieces of the various theories of learning scattered in different parts of the intelligent system and in various disciplines of science. Many of them has been developed in the areas that never exchanged the ideas and techniques earlier. Apparently they could be mutually beneficial however they have not yet been available to each other. We would like to make Engineering descriptions of learning control systems are based upon user specifications, while psychological treatment of learning processes is based upon dealing with subtle and elusive human learning activities. We want these diverse approaches and results to become consistent with each other. Philosophical thought has collected numerous insights concerned with learning processes. We would like to make this treasury of concepts available for the designer of computer architecture, or for an engineer contemplating an intelligent controller with learning capabilities.

We will use terminology that is potentially acceptable for all of the areas involved. Learning is demonstrated as a result of consecutive applying of the triplet of operations: focusing attention, grouping, combinatorial search (GFACS, see Chapter 2) to the incoming information. All existing concepts of learning are covered by this general view. Algorithmic structures described in this Chapter are intended to be instrumental for unifying diversified results from different areas dealing with learning. They strengthen the concept of multiresolutional architecture of the intelligent system, fit within the concept of multiresolutional knowledge representation, hierarchical system of goals and hierarchical behavior generation.

10.2 Definitions of Learning

It is an enormously difficult problem, to formulate anything related to the area of learning at the present time. The term *learning* has many definitions, those definitions have even larger number of interpretations. As a result, anybody using this term has good chances of being inadequately understood by different scientific communities. Let us review several definitions of *learning* existing in the literature, (including the one that we propose to satisfy at least the community of researchers working in the area of intelligent control, robotics, machine intelligence, theory of control, communication systems, and autonomous robotics.

1. Webster Dictionary: "Learning is knowledge or skill acquired by study in any field; **the process of obtaining**[1] **skill**[2] **or knowledge**" [1].

[1] and storing, isn't it? Memory is presumed.

This definition is an important one for understanding the situation: "I learned something from him" is commonly understood as follows: "He communicated to me some knowledge that I (presumably) understood[3] it, and/or memorized it. So, I learned it!" To "know something" is presumed to be always a result of "to learn something". Since we can get some knowledge by looking in a dictionary, this way of getting the knowledge is also considered "learning". In fact, there is some gigantic "underwater" part of this type of learning process. When he told me *this*, it became *knowledge* only if I trust him, e.g. as a result of many times verified experience of seeing him talking truth. Otherwise, I would suspect that what he told me might not be a truth and it won't qualify to be considered *knowledge*, I won't be able to claim that I *learned* it. So, **learning presumes getting the information validated by an acceptable level of belief**, that is information that underwent a process of belief validation. (It is tempting to state that in the process of belief validation the initial information, or an initial hypothesis, or a tentative concept, is being transformed into knowledge).

2. Encyclopedia Britannica: “Learning (animal) - **the alteration of an individual behavior as a result of experience**” [4] [2]. This definition alludes to development of reflexes, emergence of evolutionary changes related to behavior, etc. Interestingly enough, there is no mentioning of any knowledge acquisition! This definition reflects the frequent conviction among psychologists' that animals can “know” only in a form of behavior, animal's knowledge not reflected in behavior is not considered to be knowledge, since many psychologists still are trying to make a drastic distinction between cognitive processes in a human, and in an animal.

3. G. A. Kimble: “Learning is **a relatively permanent change in a behavioral potentiality**[5] that occurs as a result of reinforced practice” [6] [3]. Here, a new factor is mentioned: *behavioral potentiality.* This invokes a need in a place where this *potentiality* is stored, and the process of

[2] understood as useful operational knowledge.

[3] What does it mean? The issue of *understanding* is supposed to be one of the key issues of *intelligence*.

[4] It is tacitly understood that the knowledge of an experience is being stored.

[5] Behavioral potentiality looks like a code word for memory of the behaviors which could be exercised. A provisional definition of *behavior*: program of (purposeful) operations available.

learning leads to changes in this storage. In the meantime, the source of this potentiality is explicated: *reinforced practice*.

4. A. Newell, J.C. Shaw, H. A. Simon: "A learning situation[7] requires another program, called the learning program, that operates on the performance program as **its object[8] to produce a performance program better adapted to its task**" [4]. Apparently, speaking of *learning situation* refers to a *process of learning* that changes the storage from the previous definition. In order to conduct this process, the existence of *an algorithm of learning* is virtually postulated.

5. H. A. Simon: "Learning denotes changes in the system that are adaptive in the sense that they **enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time**" [5]. Oh, yes! One cannot talk about learning exhaustively without mentioning what is it for: for improving the creature that learns and correct its prior behavior, improving the cases of behavior in the future. We start getting used to the situation that learning is something that depends on learning process, that both presume a creature that demonstrates some behavior, that this behavior can be more or less efficient and the process of learning serves as a tool of increasing its efficiency.

6. M. Minsky: "**Learning is making useful changes in the working of our minds**" [6]. Apparently, no rigid definition is supposed to be expected within this particular source but it is important to detract attention from the behavior as an object of learning and focus upon the carrier of the acquired knowledge. The Mind is mentioned rather as a metaphor: no doubt, that the acquired knowledge can reside not necessarily in the Brain only, or even not necessarily in CNS; each subsystem, each module of the system, each parameter of the body could be carriers of the acquired knowledge…

[6] Again: something is supposed to be memorized.

[7] So, this is not a direct definition of *learning*: the learning process seems to me a more important issue than the concept of learning itself.

[8] An important loop: *learning* program is supposed to exist before the *operation* starts in order to enable this current *operation* to improve something in the future *operations*.

7. R. Michalski: "**Learning is constructing or modifying representations of what is being experienced**. ...A fundamental problem in any research on machine learning concerns the **form and method used to represent and modify the knowledge or the skills being acquired**" [9] [7]. This definition focuses on the representation only. Neither the mechanisms not the purpose of learning are addressed. The important detail: our experiences – this is what is supposed to be reflected in the representation!

8. J. P. Guilford: "**Learning is essentially discovery**[10]. ...The achieved cognition may be an acquaintance with a new unit, the formation of a new class, the formation of a new relationship or a new system, the awareness of a transformation, or the extension to new implications" [8]. This paper of 1961, demonstrates that many particular learning processes can be interpreted within this definition of learning: memorizing, skill learning, reinforcement learning, learning while problem solving, etc. The special novelty of the acquired knowledge is underlined in this definition. The results of learning are expected to be not something that could be deduced by our mind but something that emerged unexpectedly, could not be routinely foreseen, the phenomenon of "emergence" is required. The concept of "emergence" is not well defined. However, in many cases it is known to be associated with the phenomenon of generalization (see Chapter 2).

9. M. I. Shlesinger: "Learning of image recognition is a **process of changing the algorithm**[11] of image recognition **in such a way as to improve, or maximize a definite preassigned criterion characterizing the quality** of recognition process" [9]. After all previously discussed definitions, this focusing on a novel algorithm seems to be a narrowing of the class of phenomena defined a learning: indeed, a discovery of an unexpected entity within an image seems to be an "emergence" and might be qualified as learning, too.

[9] Again: memory is presumed. However, the location and organization of the storage are yet to be explicated.

[10] This does not look as definition of learning. The author was concerned with bringing about a message of importance of the phenomenon of innovation process which is associated with learning. Actually, he wants to say that just a simple memorization is not a final result of learning. The product of learning is expected to be a discovery as a result of the intermediate process of memorization.

10. Y. Tsypkin: “Under the term learning in a system, we shall consider a process of forcing the system to have a particular response to a specific input signal (action) by repeating the input signals[12] and then correcting the system externally”[13] [28]. This definition given by a prominent specialist in control theory contains remarkable conceptual bridges to biological learning, including reinforcement learning, development of reflexes, etc.

The resemblance of this definition to the principle of “Hebbian (or Pavlovian) learning” is a remarkable one. I. Pavlov discovered the process of unconditional reflex learning in 1934. In 1949, D. Hebb formulated similar statement for neurons: “When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes place in firing it, some growth process or metabolic change takes place in one or both cells such thatA's efficiency, as one of the cells firing B, is increased”. [85].

This multiplicity of definitions, the variety of angles and scope in approaching the problem of learning creates an apparent feeling of dissatisfaction, and some authors admit the need in reconsidering the existing definitions. P. S. Churchland expresses this dissatisfaction as follows:

“The general category of learning has already fragmented into a variety of kinds of of process, and indeed the term ‘learning’ is now often replaced by the broader and less theoretically burdened expression ‘plasticity’” which include “habituation, sensitization, classical conditioning, operant conditioning, imprinting, habit formation, post-tetanic potentiation, imitation, song learning (in birds), one-shot learning to avoid nausea producing foods, and cognitive mapping, in addition to which are the apparently high-level phenomena distinguished in terms of what is learned, such as learning language, learning who is a conspecific, learning to read, learning social skills, learning mathematical skills, learning to learn more efficiently, learning to lower blood pressure, and heaven knows what else” ([11], pp. 151-152).

The author chooses to characterize the situation with learning as a “pre-theoretical” situation.

Some of these definitions contain indirect reference that the learning process should incorporate information about prior operations of the system, possibly should generalize and make classifications on the information collected. All of them refer to potential changes in operation. Some of them are talking

[11] This is a remarkable statement: learning presumes a change in the algorithm of operation!

[12] testing? with subsequent storing of the results of the testing?

[13] Why not internally? The ability of intelligent systems to produce goals at a level of resolution should be bounded by the set of sub-goals for the goal which was prescribed from the level of lower resolution.

about modifying the results, one of them expects discovery from learning. All these definitions are based upon an idea that a process of collecting information about experiences can be organized in such a manner that some similarity of our interest will be discovered in this the information. Then, the results of our observations will be unified by this similarity, and the results of this grouping will later allow to change the algorithm of operation, and this change will bring an improvement!

Let us recapitulate the focal points of all above definitions. Learning should have the following features:

1. to be a process of obtaining skill (the term "skill" presumes the existence of a program that produces a useful behavior; this program as well as the corresponding knowledge should be represented somewhere)
2. to produce alteration of an individual behavior (the term "alteration" implies that the program to produce a behavior should be changed)
3. to produce change in a behavioral potentiality (the term "potentiality" alludes to a storage of the programs)
4. to develop an inner program better adapted to its task (the term "better" implies that the measurement of "goodness" should be build-in into mechanisms of learning)
5. to enable performing the task more efficiently (the term "efficiency" implies that the goodness of performance should be measured)
6. to change the quality of the output behavior (the term "quality" implies that the "goodness" of the output behavior must be evaluated)
7. making useful changes in [mind] (the term "useful" refers to "goodness"; the term "mind" alludes to the ability to store and modify the programs)
8. constructing or modifying representations (the term "constructing" implies not only an ability to store but also an ability to create new programs)
9. learning is essentially discovery (the need to create new programs is emphasized)
10. formation of new classes and categories, generalization (this formation of new programs should be achieved via generalization)
11. changing the algorithm (the term "algorithm" should be understood in the same way as the term "program" was used in the observations above)
12. forcing the system to have a particular response to a specific input signal (action) by

repeating the input signals (two concepts are introduced: the one of "response" and another of "repetition" of the experience which requires this response).

A definition which addresses most of these features was given by G. Edelman.

11. G. M. Edelman: "In an environment containing unforeseen juxtapositions of events that may affect survival, it is learning, not just perceptual categorization, that ensures successful adaptation. …Perceptual categorization and memory are therefore considered to be necessary for learning but obviously are not sufficient for it. (pp. 56-57). Learning arises as a specific linkage between category and value in terms of adaptive responses that lead to changes in behavior ([12], p. 152). The concept of "survival" can be interpreted within our vocabulary as the ultimate (and very special from the point of view of tools applied) way of evaluating the "goodness" [14].

We will synthesize a definition that blends these ideas of knowledge, and skill acquisition and/or modification, change of behavior via change in the program, efficiency and quality of operation as a goal of learning, and achieving these changes in the program as a result of discovery. This definition is formulated as to be applied for a couple: Control System-Controlled System. Dealing with this couple is adequate to a multiplicity of technological problems, and allows for powerful interpretation in a number of biological, psychological, and sociological cases.

12. *Definition:* **Learning is a process based upon experience of functioning[15] of intelligent systems (including their sensory perception, world representation, behavior generation, value judgment, communication, etc.) which provides a better value(s) of the cost-functional(s) considered to be a subset of the (externally given) assignment[16] for the Intelligent System.**

Development consists of *modification* and *restructuring*. Development can be a part of design prior to system operation, and can be a part of normal system operation. Thus, it can be said

[14] Actually, this substitution of the issue of "survival" by the issue of "goodness" is a result of our desire to avoid (or substantially reduce) references to the term Darwinism. The topic Darwinism and Learning deserves writing a separate book.

[15] By using the term "functioning" we actually are able to address all processes pertinent to intelligent systems including those of design and development (modifying and/or restructuring), planning, etc.

that learning is a perpetual redesigning of a system. Modifying can be understood as parametric adjustment of these algorithms with no changes in their structure, (e.g. making corrections in the range of the rules without changing the rules). Cost-functional is assigned by the upper level of resolution (e.g. external user, population of intelligent systems, etc.) which is assumed to be a source of the specifications for functioning.

In fact, a more complicated case can be considered when the user assigns only general policies, and the lower level cost-functionals are formulated and then, modified and restructured by the system itself. We won't talk about this type of systems in this Chapter, they seem to be a remote research goal. Our immediate goal is to discuss learning control systems which can work under preassigned cost-functional.

Restructuring presumes not only modification of existing rules but also creation of the new rules[17], or the new meta-rules. This is a more complex, more sophisticated case of learning. The highest level of learning takes place when the goals of the functioning could be reconsidered based on the results of the operation of the subsystem of learning. The following analogy can be of importance: the system with learning is permanently undergoing the on-line process of re-design. The system with no learning, is a system that has been designed in the past once and forever and does not need (or cannot afford) any improvement.

Both *learning via modifying*, and *learning via restructuring* can be based upon two types of knowledge: knowledge obtained from external source of knowledge ("learning by being told"), and knowledge created within the learning system ("learning by discovery"). External source of knowledge is evaluated by the degree of belief, and therefore it just substitutes for the internal subsystem which creates the knowledge within the learning system (and which results of operation are also evaluated by the degree of belief). Thus, the systems with learning by being told can be considered a subset of systems with learning by discovery, in which the process of discovery is externalized.

Our definition of learning can be helpful in defining learning system, or learning control system. We will try to make it step by step, and we will start with defining control systems we are going to deal with.

The ultimate learning will lead to the change in the design of the creature (which

[16] Later, we will call it a "goal set".

apparently is done within the process of evolution). Therefore, we will understand learning as a process of evolving the model of the world (including the models of sensory processing and behavior generation) for the benefit of the system under consideration.

Decision making as a process of selecting among the alternatives is not a process of learning (as soon as the decision algorithm is in place). However, development of these alternatives is linked with learning. The same can be stated about planning. To construct a plan of actions is not learning if this plan is obtained from the existing look-up table. However, planning IS LEARNING if the process of designing new rules is a part of the plan construction. This happens always when the process of search is involved and thus the result is not a combination of the previously stored components but a discovery of new models of behavior that have not been explored previously.

Subsequently the following processes employ learning: concept generation, data classification, image (object, entity) recognition. Logical schemes of all corresponding algorithms are similar.

10.3 Implicit and Explicit Logical and Psychological Schemes of Learning

10.3.1 The Need in "Bootstrap" Knowledge: Axioms and "Self-Evident" Principles

Knowledge acquisition cannot be done without existence of some initial "background" knowledge, or "bootstrap" knowledge. This "axiomatic" knowledge is a matter of faith: it cannot be learned, it should exist in the system prior to the learning process, the system should just to believe that "this is so". It is very instructive, to determine the axiomatic structure within the existing learning schemes, and to built the advanced learning theory based on a set of explicitly stated axioms. The following general properties are expected from the general truths, apriori laws, axioms [13]:

(a) unrestricted generality ("always true and true of everything"),

(b) independence of sense experience ("cannot be proved or disproved by an examination of sensory experience"),

(c) self-evidence ("cannot possibly be doubted").

Less evident principles cannot be considered axioms, especially when they are instilled by the context. We will list several such semi-axioms (SA).

[17] see "discovery" from the definition 8 of **Introduction**.

SA1. Every observation is evaluated for the degree of truthfulness, degree of belief, probability, likelihood, goodness, possibility, etc. It is presumed that the results of such an evaluation are necessary (a) to decide upon memorization, and subsequent retention of the particular observation, (b) to decide upon validity of using this particular observation for subsequent generalization, or to assign the value of belief to the results of generalization, (c) to decide upon action to be done after this observation. In all cases some **process of decision making is presumed** after the observation took place.

SA2. Every observation is presumed to be associated with prior observations of the same object, as well as with observations concerned with the associated objects.

Partially, this "bootstrap" knowledge is embodied in various incarnations of the GFACS-triplet: algorithms of clustering [86], classification [87], and approximation [88]. Together with the axioms and semi-axioms the procedures of grouping, focusing attention and combinatorial search can serve a mathematical explanation for the various *gestalt-principles*.

10.3.2 Gestalt Principles: Entity Discovering Insights

These demonstrate an effort to explicate at least some of the self-evident principles in a pre-algorithmic form. Gestalt principles demonstrate themselves as an intention to look for the wholeness [14]. They are considered to be very important in all psychological treatments of the learning processes [15, 16]. It won't be difficult to recognize within each of these principles a structure of GFS (see Chapter 2). Each of them is based upon procedures of grouping, focusing attention, and combinatorial search.

1-st Gestalt Principle: The Law of Figure-Ground Relationships

The existence of an entity is determined by the existence of the rest of the world. This means that the entity is discernible from the rest (extracted from the "chaos") in the same way as the *figure* is discernible from its *background*. This is possible due to the existence of an operation of distinguishing some unspecified (not initially discovered) properties of the entity as opposed to the properties of the background. This means that the following important facts are implicitly accepted:

1. possibly, there exist an entity (as opposed to the rest of the world which is considered

background, an amorphous environment)

2. possibly, there exist a property or a set of properties that can be perceived for each particular space (or location) at each moment of time
3. there exist a procedure of identifying properties
4. some of the properties are stable (not changing) for some locations
5. an entity can be characterized by a particular set of properties typical for this particular entity
6. there exist a procedure of grouping together "spaces" ("locations") with similar stable properties
7. there exist a procedure of characterizing the group of these locations by some meta-feature (a basis for assigning a name to this group)
8. the set of procedures above can be stored as a **procedure of identifying** the entity with its particular property (or properties).

This procedure apparently consists of the operations of focusing attention (on the particular locations), grouping (associating together in a single information structure), and searching for the locations which fit better for the subsequent grouping.

Existence of a property (or a set of properties) means that prior to learning the entity we should learn the property. If no property is known, no entity can be distinguished from the chaos. It is important also to stress the fact that the operation of distinguishing the entity from the background without prior knowledge of the entity existence, its properties, its relation to other entities, and/or similarity with other entities, can be qualified as **discovery**[18].

2-nd Gestalt Principle: The Law of Similarity

We already exercised this law when the locations with similar properties were grouped together. Entities are perceived as a group too if they are similar to each other in some way. This property of similarity is sometimes referred to as a gestalt property. "When the memory-images of successive notes are present as a simultaneous complex in consciousness, then an idea of belonging to a new category, can arise in consciousness, a unitary idea, which is connected in a manner peculiar to itself with the ideas

[18] This may be considered a definition of *discovery.*

of the complex of notes involved. The idea of this whole belongs to a new category" [16-18]. This new category is not a simple sum of the initial set of the elementary properties it is formed of: it is a quality that transcends any particular set of such elements [19, 20].

In this view the Pavlovian conditional reflexes using ideas of association, generalization, and rule memorization [21] can be easily interpreted as creation of a new category based upon similarity between experiences observed and memorized. (The issue of repetitiveness of the similar experience will be discussed later, in the sub-sections dedicated to induction and abduction).

A property of being similar (**similarity property**) is implicitly postulated here. An **operation of determinimg the property of similarity** as a relationship among a pair of entities is presumed. A procedure is available of **grouping together (clustering)** all entities coupled with each other by the relation of similarity. This cluster is claimed to form a pattern[19], i.e. to be perceived as another entity[20]. In other words, **entities gathered together based on their similarity form other entities**: entities are nested. (Aristotel: principle of similarity in establishing associations). Therefore we should agree that implicitly one more Law is referred to:

3-rd Gestalt Principle: The Law of Nesting

Specialists in Gestalt theories of learning, are referring to the hierarchy of gestalt properties, and entities which is formed as a result of nesting, as to "complections of higher order" [18]. Both, the Law of Similarity, and the Law of Nesting describe implicitly a converging process of learning the nested hierarchy of entities from the chaos. Nesting is a result of recursiveness in applying and functioning of the GFS-triplet.

4-rd Gestalt Principle: The Law of Proximity

Entities are grouped in patterns (e.g. other entities) if they are proximal to each other in space and/or time. (Aristotel: the law of contiguity). Certainly, the idea of proximity stimulates an introduction of the *distance* in a particular space of consideration, etc. So, in order to apply the Law of Proximity, the idea of distance should be learned beforehand as well as the skill of evaluating it (possibly, the idea

[19] Pattern can be defined as an entity which characteristic property has been declared or discovered.

and the skill are coming together). The idea of proximity is a property generating idea [19-21]. It states that similarity of properties is a basis for grouping the entities together. Proximity to each other is an additional property that should be taken in account in the process of cluster generation. Proximity in time is a basis for unifying changes related to each other into a single *discrete event*.

5-th Gestalt Principle: The Law of Good Figure

Among all possible patterns, *the best* one is to be selected when a pattern is being formed from a set of entities. An ability to evaluate some criterion of *goodness* is expected. So, the idea of goodness should be learned in advance. When the problem of grouping by similarity is ill-posed, and a multiplicity of solutions is expected, another (goodness) criterion is to be applied for the selection. It is hinted upon a need to consider goodness outside of the circle of the logical arguments: it should be something like beauty of solution, symmetry, simplicity, etc. [16, 20-24]. The idea of simplicity brings us to the idea of efficiency. The concept of efficiency is perceived as something like “beauty”. Similarity with prior positive experiences is also perceived as something like “beauty”.

6-th Gestalt Principle: The Minimum Principle

"We perceive the simplest of most homogeneous organization that fits a given stimulus pattern"[16]. The following procedure is presumed by this principle: the alternatives of entities are generated. Their homogeneity (uniformity of properties attached) is evaluated. The reduced set of alternatives contains only highly homogeneous candidates. This set is compared using a criterion of simplicity. The entity selected is what we should learn. An ability to know the ideas of homogeneity, and simplicity is implied as well as the skill to measure, evaluate, and compare them. If the efficiency is achieved, it is perceived as something like beauty, too.

7-th Gestalt Principle: The Law of Closure

If the pattern formed seems to be incomplete it must be complete. The concept that “something is meant to be complete” is the result of spatial and temporal predictions that are ingrained in all procedures of learning. We expect closure because we saw many closures in similar situations. One

[20] So, in fact we are again talking about *entity discovery.*

more confirmation satisfies our feeling of beauty. In these procedures, a set of patterns known from prior experiences is presumed to exist for possible comparison. The entity formed is meant to be compared with the entities from the *set of good patterns*. This means that the set of good patterns should be learned in advance [22-24].

8-th Gestalt Principle: The Law of Field

Any system can be considered a point in a multiplicity of fields, where each field generates its forces. The behavior of the system can be understood only if the whole set of forces is considered as a whole [20]. An ability to consider a multiplicity of factors simultaneously is presumed.

10.3.3 Repetitiveness: Induction, Abduction, and Deduction

Prediction (in time an in space) is an important component of decision making as well as an important component of any gestalt experience. Thus, repetitiveness is ingrained in the characterization of objects and processes we are interested in. Hence, our desire to evaluate possibility, probability, likelihood [25]. Thus, repetitiveness generates anticipation that stimulates prediction. All existing algorithms of prediction employ the phenomenon of statistical consistency (hidden within the concept of “repetitiveness”). This is why *induction* underlies all our behavior generating processes (see Chapter 2). If the experience of repetition is insufficient the risk of failed prediction increases. Nevertheless *abduction* (an induction with insufficient basis for expecting repetitiveness) is both frequent and necessary part of learning processes.

Induction as a source of concept generation, and or parameters adjustment, was linked by researchers to the theory of probability: "the theory of probability shows how far we go beyond our data in assuming that new specimens will resemble the old ones, or that the future may be regarded as proceeding uniformly with the past", says W. S. Jevons [26]. Another component of induction is a judgment of what is cause, and what is the effect. Russell substitute the principle [52]: '(a) The greater the number of cases in which a thing of the sort A has been found associated with a thing of the sort B, the more probable it is (if no cases of failure of association are known) that A is always associated with B; (b) Under the same circumstances, a sufficient number of cases of the association of A with B will make it nearly certain that A is always associated with B, and will make this general law approach certainty without limit"'[53].

Deduction is required because reasoning should be applied to arrive at conclusions from the repetitive phenomena. However, predicate calculus of the first order might be not sufficient. Some researchers envision the growing need in the predicate calculus of the second order: the one in which we do not infer statements by the virtue of predicates, rather predicates become variables and subject to reasoning about themselves [89].

10.3.4 Recursion and Iteration

When the law L is discovered, it can be applied using the laws of Recursion (L_r) and/or Iteration (L_i). Recursion is defined as a statement where the law L_r is applied to a number of variables among which there are results of the prior application of this law to a reduced number of variables. Consecutively applying L_r to the reduced number of variables, we eventually approach the "root" situation in which the immediate results of applying L_r could be obtained directly and do not require any additional computation and/or action (“converging recursion”). Most of the learning schemes appeal to recursion since the computation of L_r at each step requires prior computation of L_r related either to another resolution level, or prior computational result. (The number of levels involved as well as the number of prior computations required is assumed to be finite).

On the contrary, the computational schemes employing L_i do not require any consecutive nested process described above. The law of iteration is formulated in such a way as to appeal only to previous computation that has already been definitely performed. In comparison with recursion, the iteration requires less memory, although the symbolic representation might be more cumbersome.

10.3.5 Typology of Learning

The following different types of learning processes and schemes are visualized by specialists in psychology [22]. We reformulated and rearranged the list of types so that they fit within the subject of our book.

Type 1. **Signal Learning**, their association by similarity, their discrimination by the lack of resemblance.

Type 2. **Learning of Stimulus-Response** (S-R) couples[21] as a result of generalization upon experiences. Learning a chain of S-R couples, or events is called *chaining*. In the case of chaining, the system is learning a series of responses in a definite order.

Type 3. **Learning associations between S-R units** or "stimulus-response" couples, labeling the associations. *conditioning* is an important class of the type 3. The latter contains two subclasses: classical and instrumental conditioning.

Classical conditioning (discovered by I. Pavlov) can be illustrated by the following example: some neutral stimulus, such as a bell, is *repetitively* presented just before delivery of some effective stimulus (say, food placed in the mouth of a dog). A response (salivation) usually evoked only by the effective stimulus, eventually appears after the neutral stimulus is presented.

Instrumental conditioning is a variation of classical conditioning oriented toward obtaining a reward or avoiding a punishment. Animal may be taught to press levers for food; they also learn how to avoid electric shock.

Skill learning can be interpreted as belonging to one or both of conditioning classes.

Type 4. **Learning S-R Classes**, or clustering the “stimulus-response” couples, i.e. associating these labels (words), formation of their classes. Psychologists call it frequently “principle learning”. A subject may be shown sets of figures belonging to different classes (some of them are rectangles, other are circles). After repetitive rewards, the subject may learn to distinguish any "oddity". Formation of S-R classes is tightly related to the Type 5 “Discriminating Objects” and to the Type 7 “Concept Learning”.

Type 5. **Discriminating objects** with the same labels (later it gives an opportunity to discover new concepts). In discrimination learning the subject is reinforced to respond only to selected sensory characteristics of stimuli. Pigeons can learn to discriminate differences in colors that are indistinguishable to humans without special devices.

Type 6. **Rule learning**: the S-R couples can be inverted so that causal links could be determined (what “action” should be produced in order to achieve an “effect”).

Type 7. **Concept learning**[22]: the “stimulus-response” couples can be analyzed, and the

[21] S-R is understood as follows: what is the “measurement” (sensation, stimulus) that usually precedes some action (response).

[22] “Concept learning” from the psychological, or practical point of view should not be confused with “PAC Concept Learning” (see Type 9).

concepts can be obtained as a result of this. The concepts are discovered as "objects-actors" within the S-R couples. This type is tightly related to the Type 4. Actually, learning processes belonging to Type 4 become "concept learning" after the oddity is realized and given a label (the concept is constructed and defined).

Type 8. **Problem solving**: it should include a process of learning how to solve the problem. We will consider this type of learning a part of planning process.

Numerous new types of learning emerged recently as a result of new wave of learning research within the areas of control systems, artificial intelligence, machine learning, etc. This is a brief list of the most important of them.

Type 9. **PAC Concept Learning** is a type of learning introduced within the domain of "computational learning". PAC means "probably approximately correct" [30]. PAC is an abstract paradigm of concept learning formulated as follows. One should discover a "target concept" by taking samples from a set of concepts (with some probability measure) and receiving from an "oracle" a judgment whether the sample contains the target concept. The concept is considered PAC learnable if the accuracy of hypotheses is growing when the number of samples is growing, and the absolutely accurate hypothesis is achieved when the number of samples approaches infinity.

Type 10. **Model Free PAC Learning**: analyzing the source of information about the concept rather than deciding what to do next [31].

Type 11. **Distribution Free PAC Learning**: development of the hypotheses when there is no probability measure available [31].

Type 12. **Learning Concepts from ABI** (Attribute Based Instances) processes information of properties of single objects (concepts) and presumes deductive analysis of samples (instances) determined within the version space [32], inductive analysis [33], and/or clustering methods [34].

Type 13. **Learning Concepts from Structured Instances** which explores information of associations between the objects (concepts) for a possible cluster formation [35].

Type 14. **Learning by Complementary Discrimination** is a research exploring the fact that generalization by clustering similar samples is equivalent to generalization by discriminating from the complementary samples. This type of learning in introduced in [36] although it was admitted that exercising this approach is a common practice in almost all papers on learning when generalization is employed.

Type 15. **Learning by Testing the Environment and Generalizing** (TEG-learning) upon the results of experiences properly stored and organized is the most common practice known for the cases of early learning [37, 38]. It is explored for the domain of automata theory [39], and simulated for a variety of applications [40-42].

Type 16. **Q-Learning** is learning from information of delayed rewards [43]. This technique has the dynamic programming embedded. In other words, it relies upon the local tests of the environment, attempts envision the proper continuation of the behavior and computes the probability of rewards with taking in account the function of discount: the value of future rewards is discounted in comparison with the value of immediate rewards[23]. Cases of combining Q-learning with generalization are surveyed in [44].

All types of learning can be applied in supervised and unsupervised settings. We will be particularly interested in unsupervised learning since it is instructive for discussing intelligent systems. In unsupervised learning, all types, classes and subclasses of learning are base upon the concept of reward (positive or negative ones) and thus, all of them are reinforcement learning.

One can see that Types 9 through 16 do not add too much to the types 1 trough 8. They rather explore further decomposition of the classes 1 through 8 into subclasses based upon the concrete algorithms under various circumstances. This decomposition is actually unbounded: many circumstances can be taken in account in different combinations. In addition, the stage of learning is important: no matter whether the process of learning develops at the early stage (early learning) when there is no previously collected information, or whether the base of rules and experiences is a substantial one. The typology is illustrated in the diagram Figure 10-1.

Many terms related to learning are left out. The area of learning is in the making. Researchers working in the area of machine learning include in this area everything linked with information processing. In one of the recent surveys, the following topics are considered a part of the list of methods of learning: dynamic programming, game theory, optimal control, methods of regression, Markovian and Non-Markovian decision problems, problems of

[23] This is one of these enigmatic maxim which is never questioned. Why must future rewards be discounted?

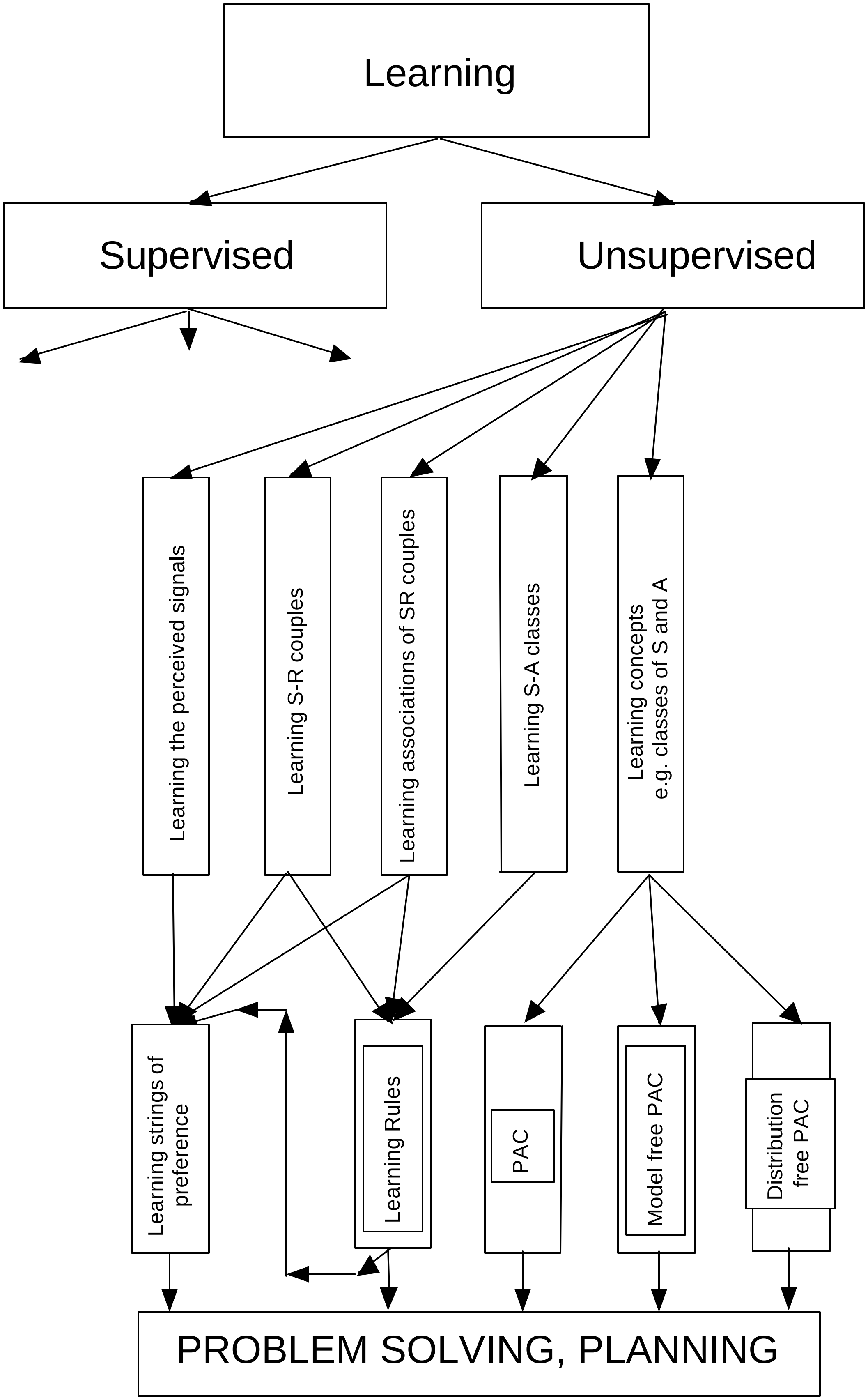


Figure 10-1. Classification of Learning

estimation and identification, theories of logical reasoning and other techniques. In [89], search methods, including genetic algorithms are considered to be methods of learning (not the tools utilized in various methods of learning). In the same source, machine learning algorithm is considered to have “examples” and “background knowledge” at the input, and gives “concept description” at the output.

We will conclude our typology with the following terms that are used in the literature on intelligent systems.

Type 17. **Repetition learning** It occurs due to repetition alone, without any feedback from the results of action. Repetition learning was discovered (implicitly) in I. Pavlov’s research on conditional reflexes. For neurons, repetition learning was first hypothesized by Hebb, and is sometimes called Hebbian learning. Repetitiveness is a part of most learning schemes when statistics (induction) is applied.

Type 18. **Reinforcement learning** incorporates feedback from the results of action.

Reward reinforcement learning in the BG system is a form of positive feedback. The more rewarding the task, the greater the probability that it will be selected again.

Punishing reinforcement, or error correcting, learning occurs when g(t) is negative, i.e. punishing. Every time the situation occurs and the punishing evaluation is given, the same synapses are weakened and the output (or its probability of occurring) is reduced.

Type 19. **Error correction learning** is actually, a subclass of the reinforcement learning. It should be considered a form of negative feedback. With each training experience, the amount of error is reduced, and hence the amount of punishment. Error correction is therefore self limiting and tends to converge toward a stable result. It produces no tendencies toward addiction.

Type 20. **Specific error correction learning** has evolved from reinforcement learning and is a subclass of the error correcting learning. **.** In specific error correction, sometimes called teacher learning, the correct or desired response of each output neuron is provided by a teacher. Teacher learning tends to converge rapidly to stable precise results because it has knowledge of the desired firing rate for each neuron. Teacher learning is always error correcting. The teacher provides the correct response, and anything different is an error.

The confusion in terminology and classification is linked with the multidisciplinary character of the issue of learning. It can be illustrated by considering AI involvement into this area. AI has been involved in the issues of Learning since its emergence (officially) in 1956 at the famous Dartmouth

Conference. However, at this moment the problem of learning control and computer learning already existed, and the broad and diversified body of literature on learning control systems, and learning automata was generating multiple research works all over the world. The body of literature dedicated to simulation of neuro-circuits and network was also growing. In our opinion, the diversity of interests and approaches in the area of learning remained as rich after AI process started as it was before. However, it would be unfair not to underline the importance of AI community unified under the codename "Machine Learning". Due to this community many important research advancements has been achieved although not in the area directly related to Learning Control Systems.

We will try to distribute Machine Learning results using the labels for the key issues[24] from [45, 46] introduced to describe the unavoidable components of the process of operation of a learning control system. We comment on the following labels referring them to the problems of the area Learning Control Systems.

a. Perception for Learning

We assume that perception for learning should be capable of providing the process of learning by knowledge satisfying some definite requirements. One of this requirements: information must be redundant (and even highly redundant). This redundancy is the source of pure learning procedures which require this *variety* (in W. R. Ashby's words [47]) which enable application of combinatorial algorithms of learning. The following topics do not reflect the actual problem of information acquisition, or reflect only very particular facets of this process.

-Acquisition of proof skills
-Learning from Observation
-Learning by Experimentation

The phenomena of generalization and nesting are not fully appreciated in AI Sections on perception.

b. Representation for Learning

Researchers in the areas of entity-relational approach of data organization has developed a powerful body of implementable results (see [48]). Mathematical background was presented in a number of papers (for example [49]). Being supplemented by results in the area of semantic networks (e.g. [50]) and conceptual modeling (e.g.[51] these results combined together, can serve as a good theoretical background for representation for learning. Still some issues are painfully underdeveloped (such as issues of similarity among linguistic and quantitative hierarchies, or descriptive and numerical space tessellations). In the literature on learning (see subsequent topical labels) these results are often ignored.

-Learning as organizing information (e.g. about physical domain)
-Learning to classify
-Goal-Oriented Classifications of Structured Objects
-Chunking the goal (goal hierarchy)

[24] It is a matter of further discussion whether the particular source of the key issues is relevant for the problem under consideration.

-Acquiring Knowledge from Information Management

On the other hand, the multiplicity of existing results on the classification theory is not utilized clearly, because those results were published in the information domain of automatic control theory. (Significance of the general theory of classification by M. Aizerman, E. Braverman, and L. Rozonoer [52-54] cannot be underestimated).

c. Storing for Learning

Learning experiences, as well as other components of learning systems refer to the need in storing the information for learning. However, the problem of storing is not specified in desirable detail. Indeed, what is the suggested organization of the storage? How long should we store? When can we forget what we remember? These and other burning questions are not addressed in the literature.

The following statements are related to the human memory (or any anthropomorphic memory) but can be fruitfully interpreted for other kinds of information storages..

1. Perception presumes storing the images and other representations (symbolic structures). Some of the images are recognized prior to the storing, some are recognized afterwards. In both cases, the process of selecting images for the subsequent storing is unclear.[25]

2. Perception is not supposed to be a veridical view of the environment, so we are not interested in the question whether the stored images represent the *truth* about the external world.

3. Perception does not include recognition as a necessary part, the latter may be considered an independent subsystem.

4. It is argued that the brain categorizes stimuli in accordance with past experiences and present needs. This categorization constitutes the basis of perception and recognition. In the constantly changing world, we prefer not to store images and other world representations but rather *procedures* that will help us to reconstruct, analyze, understand and manipulate the world. We rely not on fixed images but *reconstructions* of the past remolded in the context of the present [58].

5. Long term (lower temporal resolution, coarse) and short term (higher temporal resolution, fine) memories must be separated: "...There are obvious difficulties involved in supposing that one and the same system can accurately retain modifications of its elements and yet remain perpetually open to the reception of fresh occasions for modifications. ... we shall distribute these two functions on to different systems" [59]. One can assume that generalization and associated with it subsequent lowering of the resolution of the information stored, is an intrinsic property of the retention of the multiplicity of the information about modifications.

6. This leads to the phenomenon of *stratification* of the information stored (retained), or retaining it in a form of nested hierarchies: "...our psychical mechanism has come into being by a process of stratification: the material present in the form of memory-traces[26] being subjected from time to time to a *re-arrangement* in accordance with fresh circumstances - to a *re-transcription*. ...memory is present not once but several times over, ...it is laid down in various species of indications" [60]. One can expect that the higher the resolution of representation the lower would be retention provided by the system ("ephemeric memory at the lowest levels" [57].

[25] partly on purpose, partly randomly?

[26] or time-sequences

7. The above statement about "various species of indications" suggests that the nested hierarchical representation can be based upon more than one principle of organization of information, e. g. based upon different modalities of sensing, and it this sense it would be prudent to expect a redundant system of (nested hierarchical) representation[27].

8. Thus, a mechanism of categorizing (mechanism of clustering, grouping, classification mechanism, taxonomic technique) based upon finding similarities (or dissimilarities) can be considered to be an important tool of preparation for the subsequent information storing.

9. Mechanism of categorizing with subsequent use in the system of representation has a resemblance with the Darwinian theory of natural selection which is based on a *principle of determining and retaining* properties (information in an appropriate form) or *principle of natural selection* of these properties (the same) which work better for survival. Existence of this mechanism on the molecular genetic levels well on the level of neuronal grouping has been proven experimentally [61-64].

10. In fact, the experiments with Pavlovian conditional reflexes demonstrate that only generalizations upon series of similar experiments are retained, although the ephemeral memories of each particular instantiation might also exist.

d. Comparing for Learning

Comparison is a key procedure in learning. More precisely, the basic operation is a search for similarity in order either to unify entities in a particular class or to determine the value and/or the character of changes. The topics “Learning by Analogy” and “Learning from Examples” refer to "analogy" and "examples" as standards for comparison. The substance of search for similarity is not propagated in an explicit manner. It would be instrumental to have these topics aligned within the structure given above for the Learning Control Systems.

e. Evaluation for Learning

Numerical evaluation of the objects, relations and event of the world is a part of adequate representation, and is a subject of learning at a corresponding level of resolution. (For more details, see the subsection 10.3.6). Comparison of hypotheses requires introduction of the relevant metrics which depend on the nature of the particular method of hypotheses generation.

f. Response Generation (Decision Making) as a part Learning

Decision Making is rather the most interesting topic in the theory of learning systems. General Problem

[27] These representations are known (in psychology of brain) as maps. Brain maps of sensory-motor stimuli were discovered in 1870 by G. T. Fritsch and E. Hitzig in Germany. Complexity and variation of sensory maps in monkeys were analyzed by M. Merzenich and his collaborators at the University of California at Can Francisco in 1983.

[28] An interesting (but highly complicated problem) can be formulated of learning the modalities of sensors. We do not consider this problem here.

Solver (GPS), and a broad variety of decision recommending production systems (including SOAR) is becoming a commonplace in the AI domain where it is associated with the following topics:

-Transformation of advice into heuristic search
-Learning as knowledge compilation
-Acquiring heuristics
-Constructing a production system
-Learning by augmenting rules
-Learning from Discovery
-Discovery and Heuristics

In fact, the major ideas have not changed since the initial GPS (1959 [4]). Learning in this structure is performed by augmenting rules [55], however P. Winston's research paradigm is far from the ideas of automated learning. The initial scheme of the automated theory formation by S. Amarel (1961 [56]) where the "discovery" should emerge, was implying a definite combinatorial synthesis of solutions for subsequent selection. However, introduction and use of heuristics (though reflected in a massive literature stream) is not substantiated by a theoretical analysis and is not attempted to be automated.

It is interesting to mention that AI treats learning processes in terms of applying different logical structures for the inference processes: deduction, induction, abduction, etc. It seems that in this treatment the key feature of learning: generalization over the redundant information sets (collected prior to the system operation) is being blurred (see subsection “h”).

g. Actuation as a part of Learning

Actuation is not only a tool for producing diversifying alternatives of behavior: it il also a tool of diversifying the capabilities of researching the behavior within a particular environment. Indeed, the more flexible your ability to produce the behavior is the broader will be the palette of responses and the more adequate model can be obtained as a result. Certainly, the resolution and dimensionality of the vocabulary of actuation should correspond to the concrete level of resolution for which the model is supposed to be obtained. The correspondence between the values of resolution of actuation and of the model to be produced is a matter of further research.

h. Generalization for Learning

Results of this analysis are applicable in the prior 7 topical clusters. Generalization is performed within the subset of information determined by the scope of attention. Within this scope, a search for similar entities is performed that can form a *unifiable* group of similar entities. A variety of combinations should be explored before the best grouping is achieved. The groups are hypotheses for the future evaluation as alternatives of S-R couples, rules, or particular concepts. The spatial and/or temporal similarity cluster is a basis for declaring (or discovering) unnoticed earlier regularity (or discovering a new concept). A variety of logical methods is used for arriving to the conclusion that the new cluster is a new concept. Clustering by focusing attention, grouping and combinatorial search supported by the available techniques (inductive, deductive, abductive, and so on) is called generalization. The list of associated topics from the literature follows, these are represented in the variety of papers dedicated to particular research instantiations.

-Mechanisms of generalization
-Conceptual clustering
-Inductive Learning
-Learning to predict sequences
-The effect of noise in concept learning
-Learning concepts by asking questions
-Using derivational histories
-Search for regularity
-Generalizing Plans from Past Experience
-Generalizations upon rich memory
-Concept formation by incremental analogical reasoning
-Theory formation

However, the literature in this area cannot satisfy the fundamental questions that naturally emerge: what is in common in all methods of generalization? Can we learn something from a non-generalizable information? Can "generalization" and "concept creation" be equalized? Can the system of multiresolutional world representation be built by consecutive applying the algorithms of generalization?

The problem starts at the stage of focusing attention. Indeed, we do not know how to select the body of information which will be used for the subsequent generalization. Apparently, this preliminary “filtering” of the available information should be done as a result of trade-off between the danger to lose the correct result by cutting off the "good” information, and to lose efficiency by processing unnecessary excessive information.

We believe that all generalization processes are based on a set of primitives of recognizing similarities/dissimilarities and are built-in within every known system of learning. This fact makes determines that the construction of the world model boils down to generalization upon the available information. This makes easier the process of computer simulation of learning processes since it becomes independent from the logical concept entertained in the particular computational system.

Level of abstraction is a commonplace in the literature on artificial intelligence. In fact, generalization always is considered when the level of abstraction is mentioned. The time came to bring some order in the situation: to determine the degree of equivalence between generalization and concept formation, to determine the degree of equivalence between abstraction and concept formation, to determine induction and combinatorial search as the only mechanisms of generalization, to accept that recursive application of the mechanisms of generalization leads to generation of multiresolutional world representations.

10.3.6 On the difference between adaptive and learning systems.

We already saw that as a result of learning, new numerical data can be acquired, and the system will "tune itself up" in the changing environment. This is *quantitative*, or *parametrical learning* which is usually associated with the domain of *adaptive systems*. This leads us to ***parametric learning controllers***, or ***parametric (adaptive) controllers.*** Another situation leads to generalizations that reject further "tuning the system up" and call for a restructuring a) the model of system, or even b) the model of controller. We saw that this process is done consistently in a model where numerical nested hierarchical processes of generalization are supplemented by *cognitive learning*, or linguistic nested hierarchical processes (which can be understood as an extension of numerical generalization processes at low resolution of representation). Therefore, these controllers are named here ***cognitive learning controllers***.

One can argue the need in introducing the term *learning* on the basis that we can use the terms *structural adaptive controllers*, or *cognitive adaptive controllers*. We do not see any need in avoiding the term *learning*. Moreover, we believe that using the term we allow for further systems development: learning control can be beneficial for any particular engineering domain. We should be interested in cross-fertilization between control and AI areas. Machine Learning, as a branch of AI exists and has demonstrated a vast multiplicity of interesting and important results which should not be underestimated, neglected, or necessarily translated into another scientific language. Specialists in control should admit existence of this scientific body: *machine learning*, and should incorporate its results in a form which would be productive and harmonious. This process was never painless, however we have plenty of evidences that this process was always scientifically fruitful.

10.4 Information Acquisition via Learning: Domains of Application

10.4.1 Estimation and Recognition as Components of Learning

The essence of learning can be exemplified by the processes of recognition and estimation characteristic for all stages of learning. In this section, we would like to assert that dealing with acquired information including numerous estimation methods is based on a specific *logical scheme* that can be visualized as follows:

- information should be acquired repeatedly about the same zone in the state space (e.g. the same object of the world); the repetitiveness should be characterized by a particular time period of sampling which pertains to a particular level of resolution,
- this information should be stored (memorized) for the subsequent dealing with the whole set,
- a law of clustering is assumed for the stored information about the same object of the world,
- if the information is not known to belong to the same zone in the time-state-space (object of the world) then existence of this law of clustering is assumed to be an evidence of belonging to the same object of the world,
- all known estimation procedures use recursive processes of computation with subsequent increasing of resolution of information used and produced,
- the results of prior estimation steps are assumed to be considered good reason for expecting similar results in the continuation of information acquisition, so, the idea of "predictors" is employed in a number of estimation methods.
- the groups found as a result of clustering should be considered a single objects and the multiple

information related to the cluster should be generalized (when numerical data is involved, the multiple data might be averaged, or substituted by their probabilistic expectation, etc.), this concludes the process of estimation.

--the generalization upon the cluster might be compared with similar cases in the past for which an interpretation was constructed (and confirmed); this conclude the process of recognition

Information acquisition is a process of receiving the **samples** of data from the sensors and transforming them into the form that will be understandable for the subsystems that require this information. Samples of sensory data $\mathbf{x_i(a_i(O), v)}$ which carry measured or assessed values **v** (a number of units of a particular scale) of an attribute **a** (name of characteristic feature of an object $\mathbf{O_j}$, quality which determines the physical units of the scale) may be represented as propositions of the form [65]: **the i-th attribute of the j-th object is value**, or (attr-obj); value), for example: [temperature (cylinder, coordinate), 86 °C).

Here, the *attribute* is determined by a modality of sensing (temperature), i.e. by a feature which is being measured by this particular sensor. Object is the entity of the external reality this feature is known to be attributed to (cylinder, coordinate). The modality of sensing (in other words, the physical essence of the particular sensor based on the nature of energy transformation process performed by the sensor) is known since the preliminary design of the machine to be controlled is known.[28] Object is often known (in the generally applied automated machines) however in the intelligent machines, the object is to be recognized, this is a part of the overall learning process. The value is a scalar quantity of an n-tuple quantified over some numerical or linguistic scale that pertains to the resolution level under consideration.

There are two types of sets for $\mathbf{x_i}$: space related sets $\mathbf{x_i(x_k)}$, k=1,2,...,m, k i, m is the total number of attributes reflected in the sensory data , and time related sequences $\mathbf{x_i(t)}$, and/or $\mathbf{x_i(x_k,t)}$. These sets and sequences contain a number of patterns $\mathbf{\{P_m\}}$, (µ=1,...,m, a number of patterns which can be recognized), which are inclusive of a set of objects $\mathbf{\{O_j\}}$ existing in the world $\mathbf{\{P_m\}}$**É**$\mathbf{\{O_j\}}$. Recognition mapping $\mathbf{C_1:\{x_i(a_i(O),v)\}}$**®**$\mathbf{\{P_m\}}$ is known to usually precede the interpretation mapping $\mathbf{C_2}$**:** $\mathbf{\{P_m\}}$**®**$\mathbf{\{O_j\}}$. Together, these two mappings (recognition and interpretation) constitute an operator of cognition **C**=$\mathbf{C_1* C_2}$. When the set of objects (entities) of the world is found $\mathbf{(C_2)}$ and the relationships upon this set are determined, the problem of control can be attempted.

We do not consider here the mechanism of perception that is based on the recognition and interpretation mappings. However, we would like to emphasize the following general two alternative rules characteristic for the overall **C**-operator: 1) the vocabulary of the output of **C** should contain only those words (and relationships among them) which are understandable for the control system, 2) otherwise it should contain a request for establishing new words with the subsequent incorporation of these new words into the operation of the control system. Both rules do not change anything in the content of the input vocabulary $\mathbf{\{x_i(a_i(O), v)\}}$**.** However since the set of **{O}** is subject to change we will denote the input $\{x_i(a_i(H),v)\}$ where the set of **H** is understood as a set of all hypotheses generated about the objects of the world.

Consider observed input $[\{x_{is}(a_i(H),v_{iks})\}, \{H\}=0]$ ® $\{x_i(x_k),v_{iks}\}$, where **s** is the number of samples s=1,..., N**.** It is assumed that there exist a *true* value of the input variable v^*_{iks} and the deviation of each observation from the true value will be called error of observation

$$e_{iks}=v_{iks}-v^*_{iks}$$

which gives the following models for judging (estimating) the value of the input variable:

Linear Model (LM) estimation [66]:

$$\mathrm{est}\ (v_{iks})= av_{iks}+e_{iks}$$

where

$$\Sigma^N[\mathrm{est}(v_{iks})].v_{iks}$$

$$a=\text{¾ ¾ ¾ ¾ ¾ ¾ ¾ ¾ ¾ ¾}$$

$$\Sigma^N(v_{iks})^2$$

b) Maximum A Posteriori Probability (MAP) estimation

$$p[\{\mathrm{est}(v_{iks})\}\text{½}a]p(a)$$

a provides for **max{ ¾ ¾ ¾ ¾ ¾ ¾ ¾ ¾ ¾ ¾ }**

$$p\{\mathrm{est}(v_{iks})\}$$

where is the probability density function (pdf) of the input samples.

c) Maximum Likelihood (ML) estimation

If the a priori pdf is unknown, then it can be assumed, for example in a form of likelihood function, and the rule of ML estimation is formulated

a provides $\max\{N/2 \ln 2p + N \ln s_e + 1/2\, s^2_e\ \Sigma^N[\mathrm{est}\ (v_{iks})-av_{iks}]^2\}$

Each of this postulates of estimation presumes that a number of measures should be executed, their results should be memorized and based upon the set of these memorized values a judgment about the true value is made which is called an estimate. The following convention is being established concerning the process of information acquisition

10.4.2 Domains of Application for the Theory

Control Systems emerged as a discovery of the practical importance of the theoretical fact that our (design) decisions can be considered a reflection of the relationships existing within representations. Intelligent Systems emerged as a response to the virtual stochastic character of the world in which non-intelligent techniques limit the opportunity to receive competitive levels of efficiency. In other words, organizing our *knowledge representation* we can discover relationships within the representation which can be expected to exist within the reality.

Early stages of the control theory were aiming toward "regulation" and stable operation of machines, and toward providing stable and accurate operation of dynamical systems with feedback, and eventually toward providing required functioning of any pre-assigned object of control, or *plant* including machines in a variety of environments, man-machine systems, and even human teams, economical systems, etc. There was no doubts that a model of the plant should have been put as a contributing part for the control system design. So, the *knowledge* of the system (the *plant*), was presumed.

Another thing presumed at the early stages was that control system should have been designed in advance in full. At least, the structure was expected to be known, i.e. obtained from the process of control system design (with the major parameters of this structure) before the controller is built and functioning of the control system started. Some of the parameters were allowed to be determined imprecisely and were subject to be constantly updated and corrected within the so called *adaptive* control systems. Gradually, control specialists discovered that there is a substantial difference between the two groups of solutions: on-line control solutions, and off-line control solutions; they can be combined but the component: *on-line control* should always remain.

Conventional control was dealing with a broad variety of problems in a spectrum of devices starting with a speed regulators in the early steam engines, and ending with stabilizing a goal oriented group of spacecrafts. However, the following problems are unequivocally considered to be difficult for solving in the paradigm of conventional control theory with no intelligence and no learning:

a) Optimum control of nonlinear systems does not have a well organized solution, because models of the nonlinear systems are traditionally inconvenient for using the established paradigm which are created for linear systems only; nonlinear control theory is perpetually in its embryonic state, and optimum control solutions cannot typically be found for important nonlinear systems.

b) Optimum control of stochastic systems can be addressed only within a simulation paradigm only, because the models of stochastic systems presume knowledge of probabilistic parameters and characteristics of the systems which cannot be provided in advance in the existing practice of design.

c) Control of multilink manipulators (either 6-DOF, or the redundant[29] ones) is performed only in an approximate manner, because the models of *plants* turn out to be so huge, so unencompassible

[29] The word "redundant" means allowing for infinite number of the optimum trajectories, containing more degrees of freedom than is necessary for performing the assignment.

that even the off-line solutions can make a predicament to a control engineer, not to talk about the on-line control which in fact is required.

d) Control of redundant systems leads to so called "ill-posed" problems; in order to solve them one has to introduce a regularizing functional which is usually done based upon wishful assumptions rather than on information known about the system.

e) Control of autonomous robots does not allow for using any of conventional techniques, because most of the information to be taken in account is not known at the beginning; thus, it cannot be supported by off-line solutions, it requires the on-line interpretation.

f) Control of systems with multi-sensor feedback information cannot be done just by using standard multi-variable approaches, because the multi-sensor information must initially be conceptually integrated which means that some generalization activities are presumed.

g) On-line control of systems with incomplete initial knowledge of the model, and/or of the environment is not supported by any of conventional techniques because we do not know how to incorporate new knowledge of the world within the model of the plant, and/or the model of the controller.

In all of these problems, neither the knowledge assumed at the stage of design could be considered complete and satisfactory, nor the process of design could be completed unless new knowledge would be additionally supplied. An opportunity to view all these system in a different, unconventional way as systems with never ending design stage, has appeared as a result of the broad application of computers, and in particular, industrial computer systems equipped by a variety of transducers-sensors.

Indeed, all of the systems mentioned in the above list, could be presented as a loop which contains the all means required for dealing with the complicated problems: *perception* for organizing a diversified information set coming from a multiplicity of sensors, *cognition* for enabling the system not only to interpret the results of perception but also to put them in a perspective necessary for determining strategies, and policies of future operation as well as to submit all necessary information for planning/control processes, *planner/controller* to generate proper control sequences, and so on (Figure 1,a)[30]. This system is a result of a finite design process shown as a mapping of the "controlled system" part into the "control system" part.

10.5 Axiomatic Theory of Learning Control Systems

In order to introduce the learning control system in the axiomatic fashion the following

[30] A question can be raised how does this system operate. Each subsystem is considered an independent entity with its own knowledge base. Of course, the knowledge bases of all subsystems can be organized in a network and allow for communication, negotiation of mutual consistency of operations, etc. Using feedback loops can be expected from cognition to perception, from planning/control to cognition, and so on (see [57]).

preliminary definitions are required.

10.5.1 Definitions

***Definition 1**. Representation Set*

Any reality of the World can be represented[31] (or modeled) by a representation set which is understood as a list of characterizations for the entities (both with their qualitative and quantitative characterizations) composing this reality, and a list of relationships among these entities[32]. (This definition is a recursive one, and it can be applied to the word *entity* within the definition; the words World, Reality, and Entity are not defined, common thesaural definitions can be applied). So the label for the entity "position" does not belong to the representation set but the statement 'position is six" does belong to it. Therefore, the lists consist of couples "entity→characterization", and entity→entity". Any desire to represent the world requires memory (storage of the representation). Each entity of the world is a carrier of its own information and could be considered a "memory of itself". This is not the kind of memory which is embodied within the representation. **Representation presumes memory of the object separated from this object[33].**

Transformations of the representation sets (of entities, and/or of their relationships) are presumed which can transform the list in a form of, for example, difference, and then differential equations (transformations are expected to be based upon assumptions that are part of the initial lists). Any consistent result of the transformation of the initial lists is still considered to be a representation set. All lists are considered to be open: they allow for endless growth, so when new labels appear they are added in corresponding lists of entities, relationships, and generate their own new lists of decomposition. Since consecutive decomposition is presumed, this type of relational structure is identical to a tree structure. Representation set is considered to be applicable only to a particular moment in time.

The recursive character of the definition for the representation set implies decomposition of each entity in its parts with the set of relationships among them, an so forth. This decomposition focuses on parts of the initial entity which require higher resolution for their consideration than can be provided within the initial representation set. Thus, any representation set is a multiresolutional hierarchy by definition (for details on the representation set see [67], subsections IV.E through IV.J on D-Structure). Time representation at each level of the representation set has also different resolution. Each representation set related to a particular moment in time is a *state* of the representation set[34] (so, a state is the snapshot of the representation set at a definite resolution level). The consecutive string of states is called a *process*.

***Definition 2**. Control System (or Controller)*

[31] or modeled: we will use the words *representation* and *model* intermittently.

[32]Unfortunately, we cannot talk about the World referring to it in another way rather than use words such as *objects*, or *entities*, and *relationships* among them. Sure, all these entities might turn out to be just our imagination, a result of our desire to superimpose some compulsory organization upon the free and chaotic world. However, this is the only way we can talk about the world. What I name here *reality* is just a meta-representation of it.

[33]as opposed to, say, memory distributed within the object.

[34]event?

Control System is a set of interconnected devices operating in such a way as to provide behavior which ends after satisfaction of a *goal set*, or a *set of output specifications* formulated for a Controlled System, and judged by a user[35] . Control system can be represented as a set of logical statements (of existence, and of belonging to a class) which can be made about the implications that hold among units of the *representation set of the Control System*. (For example: if the representation set contains the subsets of parameters and variables of a dynamic system, then the set of logical statements can be represented as a system of differential equations for this dynamic system, or the system of difference equations when the variables are tessellated, or as a list of implications if these difference equations are written in a linguistic form)[36]. Sometimes people are talking about so called control problem (see ***Definition 7***). Control problem is how to built the Control System.

Definition 3. *Controlled System (or Plant)*

The system of interest for a user which is expected to be controlled by a Control System in such a way as to provide behaviot that ends with the satisfaction of a *goal set*, or a *set of output specifications* formulated and judged by a user. Controlled system can be represented as a set of logical statements which can be made about the implications that hold among the components of the *representation set for the Controlled System*. (The example from the definition of the control system applies).

Definition 4*. Goal Set, and Output Specifications.*

A subset of the representation set called the goal set, or a set of output specifications is a list of logical statements which can be made about the implications that hold among the subsets of the qualitative, and/or quantitative *set of observables* (e.g. the *input subset* and the *output subset* as well as the set of *inner states* that allow for measuring) of the system which are accepted by a user to be characteristic for the controlled system, and which must be satisfied as a result of using the control system. (As well as the rest of representation sets of the system, goals are also nested hierarchies).

Definition 5*. Observables*

A subset of the representation set called the set of observables is a set of all characteristics (variables) that can be observed, i.e. measured (with a definite accuracy), and *recorded*. The set of observables contains (not necessarily consists of) three major subsets: the input subset, the output subset, and the set of *inner states* that allow for measuring. The input subset includes those observables that can be assigned and executed by the user's external means, the output subset includes those observables which cannot be executed by external means, which can be executed only by the controlled system, while the set of inner states includes those variables which are the components of the existing model of the plant.

Definition 7. *Control Problem*

The standard problem of control is defined as follows: for each event from the set of initial events $(t0,x_O)$ the control should be determined u(•) which transforms this initial event into the goal set, and minimizes

[35] Goal Set presumes a user. No technological problem can be understood with no user. So, all Controllers have a Goal Set. Problems are not considered here which can appear outside the technological domain.

[36] i.e. statements of existence, statements of belonging to a class, and statements of implication; mathematical statements are the same, just written in a different notation.

the cost functional simultaneously [68]. In fact, this definition is a tautological one: solution for the control problem is a control system defined earlier (see ***Definition 2***).

Another definition of a control problem has a clear reference to the *conventional control theory* [69]. The control problem is recommended to be divided into the following steps: 1) establishing of a set of performance, 2) writing down the performance specifications, 3) formulating a model of the system in a form of a set of differential equations, 4)*using conventional control theory* find the performance of the original system, and if it does not satisfy the list of requirements, then *cascade or feedback compensation must be added to improve the response*. 5) using modern control theory approach assign the entire eigenstructure, or the necessary structure is to be designed to minimize the specified performance index (which is understood a quadratic performance index). In general, one can easily find that there is a surprising lack of uniformity in the existing views on *control problem*.

The following definition of control law can be considered consistent with the practice of control (G. Saridis, [70]): **control of a process implies driving the process to effectively attain a pre-specified goal**. One can see that the notion of goal is included in this definition. Cost-functions and cost-functionals are considered to be a part of goal set. In a multiresolutional setting we should talk about a hierarchy of goals and a hierarchy of cost-functionals.

Thus, the recommended steps for solving of a control problem can be reformulated as follows. The control problem is recommended here to be divided into the following steps: 1) establishing of a set of cost-functionals for performance evaluation, 2) writing down the performance specifications, 3) formulating a model of the system , 4) finding the performance of the original system, and if it does not satisfy the list of requirements, then determining the input which is required to achieve the performance specifications with minimizing and/or maximizing the cost functionals in the sequence which make sense for the user (programmed control). 5) evaluate how the unexpected factors and quantitative uncertainties can affect the operation of the open loop system, and introduce the feed-forward and the feedback controls as required.

Our definition of a system will differ from the classical definition from [71] by introduction of a controller as a part of the system. The system is understood as a couple $\{\Sigma, \Delta\}$ consisting of the descriptive set characterized by the vector of evaluations Σ and the vector Δ of corresponding values of the distinguishability zones (of the values of accuracy). In all subsequent presentations, any variable is presumed to be assigned as a couple of its evaluation (e.g. average value) and its accuracy which can be considered as a source of information for determining a corresponding distinguishability zone (or the size of the corresponding tessellatum[37]). For simplicity we will omit the couple notation when possible, however the concept of accuracy for each component of Σ will be invoked as the need emerges. The following notations will be used: **T** -for the time set, **X**-for the input set, **U** - for the set of instantaneous input values, **W** - for the set of acceptable inputs, **Y** – for the set of outputs, **G** – for the set of instantaneous output values, **j** - for the state transition function, **h** – for the output function, **K** – for the control law exercised in the system.

[37] This means that the multiple sampling took place prior to discussion (or should be performed) within some scope (size of window of attention). This multiplicity of samples has been generalized by applying procedures of focusing attention, combinatorial search, and grouping. As a result, both the evaluation and the accuracy should have been determined.

***Definition 8**. Observed System*

A system {Σ , Δ} is a representation structure formed over nine-tuple representation sets[38] (T, X, U, Ω, Y, Θ, φ, η, K) defined by the following axioms:

(1) (*Existence*) There exist a given **time set T**, a **state set X**, a **set of instantaneous input values U**, a nonempty set of acceptable **input functions**, (or **command sequence,** or **control string**) generated by the control law **K**

$$\Omega=\{\omega : K \rightarrow U\}$$

a set of **instantaneous output values Y**

Q= {y**:T→Y**}

and an **output function**

h = T´ X→Y.

(2) (Direction of time) T is an ordered subset of the reals.

(3) (Organization)There exist a **state-transition function** (or trajectory of motion, or solution curve)

j : T ´T´ X´ W →K→ X

whose value is the state **x(t) Î X** resulting at time **t Î T** from the initial state (or event) **x_0=x(t)** at the initial time **t Î T** under the action of the input **w Î W**

Usually, the control law K is pre-determined: it is either a time-dependent program (open loop control, feedforward control), or a mapping from the set of variables (closed loop control, feedback control), or both (OLC/CLC-control, combined feedforward-feedback control

In the reality of learning control, we are interested in defining the state-transition function *before* the definite moment in time (t ≥τ): the problems of predicting are solved by analyzing behavior of the system at the interval Tm. The axioms (1) through (3) from ***Definition 8*** implies that the time scale of the system (at a level of resolution) should be pre-selected.

This set of definitions and axioms is usually supplemented by a demand of stationarity, linearity, and smoothness if using representation is expected in a form of differential equations. The requirement of smoothness determines the acceptable granularity (the law of tessellation). Properties of the state transition function lead to the following theorem proven in [71]. Every system Δ with a state transition function defined as is shown above, and with the norm $||\omega||=\sup||u(t)||$ has a transition function **at a level of resolution** in a form

$$\frac{dx}{dt} = f(t, x, \pi_t\omega),$$

where operator π_t is a mapping Ω→U is derived from $\omega\rightarrow u(t)=\omega(t)$. The suggestion to select π_t in a form π_t: $\omega\rightarrow(u(t), u'(t),...,u^{(n)}(t))$ (by L. Zadeh, C. Desoer, [72]) is rejected which furtherly narrows down the domain of systems under consideration. In the case of a smooth, linear, finite dimension case,

[38] which are nested multiresolutional hierarchies as all other representation sets in this Section.

[39] Everything said to exist is meant to *exist* in the **reality**. Existence in **memory** and availability of the entity of knowledge to be utilized for control functions should be discussed in specific detail.

the transition function obeys the simplified relations. The simplification is determined by selecting norms of the corresponding spaces with no derivatives of the time functions for controls. Only now we are coming to realize that the expectation of L. Zadeh and C. Desoer might be quite right, we are dealing now with the systems which need a norm based upon all set of u(t), u'(t),...,u(n)(t)). The complete representation of the system can be written as follows

$$\frac{dx}{dt} = A(t)\, x+ B(t)\, u(t),$$

$$y = C(t)\, x(t) + D(t)\, u(t),$$

where A(t) and B(t) are parts of the expression f(t,x,u(t))= A(t)x+B(t) u(t), C(t) is a mapping which is obtained from the equation

$$y(t)=\eta(t,x(t))=C(t)x(t),$$

$T=R_1$, and X,U are normed spaces, A(t) is a mapping A: T→{n × n matrices}, B(t) is a mapping B:T→{n × m matrices}, n is a dimensionality of states $x \in R^n$, m is a dimensionality of controls $u \in R^m$, p is a dimensionality of outputs $y \in R^p$. Proper adjustments and modifications are being made for a variety of cases: discrete systems, systems with non-varying parameters, etc.

We can see that the familiar forms for the system representation are not affected by using the paradigm determined by definitions 1 through 7 which means that the theory of Learning Control Systems can be constructed using these familiar form however having in mind their renewed meaning.

Now we can refine the Definition 2.

Definition 2* (revised) Control System (Controller).

Control System (see components of **S**) is a subsystem **K(Q, G)** of the system under consideration which generates the input according to the definite (required) control law.

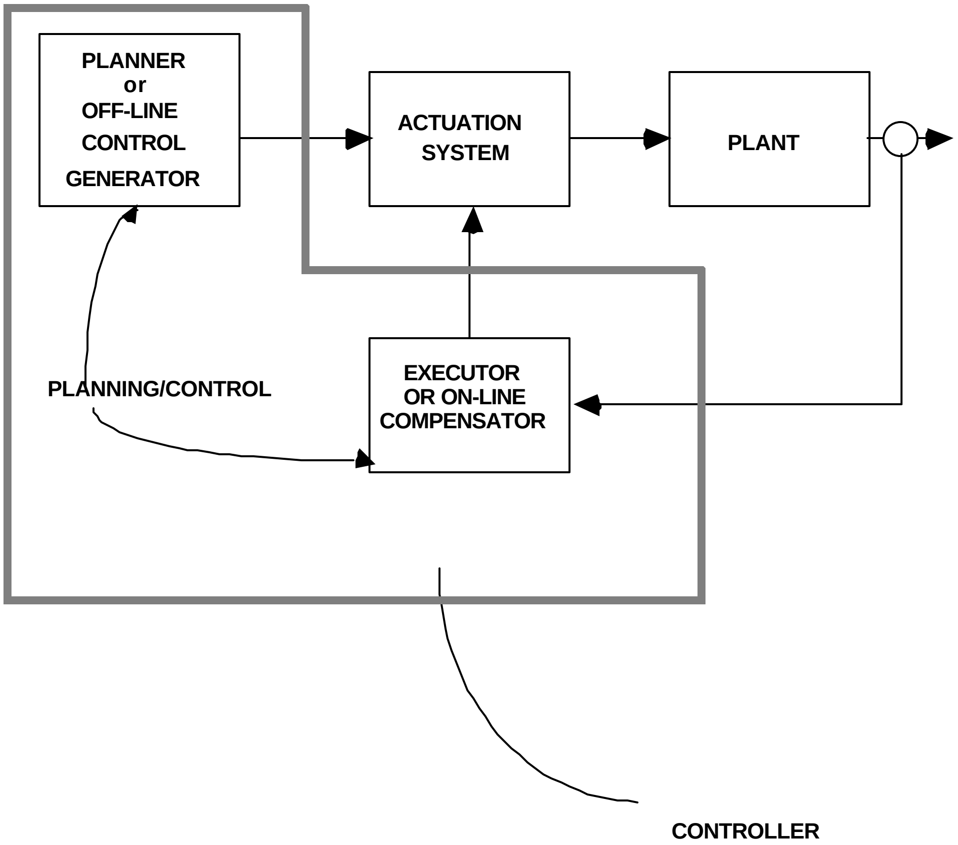


Figure 10-2 Control System

The **control law** in conventional control is defined as a permanent mapping **T ´ (X´ Y´ Y*)® U** which puts in correspondence **x(t), y(t)** and **u(t)** for each moment of time, and thus determine the a) the operator **G** of the open-loop control subsystem (which provides **U® Y, Y-Y*=0**, i.e. operation of system under assumption that no deviations of conditions and parameters, no uncertainties can exist, see the matrix **G(t)** later), 2) the operator of the feedback (**F**) and feed-forward (**D**) control subsystems (which are supposed to compensate the uncertainties entailed by applying the open loop control law). Control command sequences should be computed on-line (primarily matrices **D** and **F**), and/or off-line (primarily the matrix **G**) as to satisfy the specified list of control requirements. Control is computed according to the following equations

$$u(t) = F(t)\, y(t) + G(t)\, v(t)$$

where v(t) is a program of operation which is supposed to provide U→Y, Y-Y*=0 if no deviations from the expected conditions of operation are observed. Control System is shown in Figure 10-2)

Definition 9. Learning Control System.

Learning Control System **K[Q, $\mathbf{G_L}$(t)]** is a control system (see ***Definition 8***) which generates the input according to the control law which is constantly redefined[40] and recomputed[41] so

[40] which means that the goal set is reconsidered

that the goals **$G_L(t)$** evolving in time could be achieved and the requirements concerning the cost-functionals (which are a part of the goal set) could be better satisfied.

(We could even say that the control system is constantly redesigned on-line but in this case we have to define what is design, and this is a very complicated matter because of multiple meanings this word have in a variety of areas).

The **learning control law** is defined as a variable mapping $T \times (X\times Y\times Y^*)\rightarrow U$ which puts in correspondence x(t), y(t) and u(t) for each moment of time, and thus determine the a) the operator G of the open-loop control subsystem (which provides $U\rightarrow Y$, $Y-Y^*=0$, i.e. operation of system under assumption that no deviations of conditions and parameters, no uncertainties can exist), 2) operators of the feedback (**F**) and feed-forward (**D**) control subsystems (which are supposed to compensate the uncertainties entailed by applying the open loop control law). Control command sequences should be computed on-line (primarily, for **F** and **D**), and/or off-line (primarily, for **G**) as to satisfy the specified list of control requirements. Control is computed according to the following equations

$$u(t) = F(t)\, y(t) + G(t)\, v(t)$$

where v(t) is a program of operation which is supposed to provide $U\rightarrow Y$, ($Y-Y^*=0$ if no deviations from the expected conditions of operation are observed). All operators are being reconsidered on a regular basis, and the algorithms of their change are introduced. These algorithms are called algorithms of learning, and they are considered to be a part of overall Learning Control System (see Figure 3).

10.5.2 Learning Control Systems

The axiomatic introduction was undertaken with intention to receive principles of learning implicitly from the need to learn as it emerges in the control systems. Indeed, since the knowledge of the system cannot be considered exhaustive, or presented in the adequate form, since the knowledge of the EXO-system (see subsection 1.5) is rudimentary in the vast multiplicity of cases, since the goal-set is subjected to modification, and evolution, to change in general, a subsystem is required which takes care of all these changes and keeps the operation of the system consistent with the goal-set.

In Figure 2 the feedback compensator is introduced which adjust the control commands to the unpredictably changing circumstances of the operational environment. This compensator demonstrates one of the greatest commonsense principles ever utilized in technology: the feedback principle. Feedback compensation functions as a rudimentary system of learning: this feedback exercises the **principle of reflex** which characterizes actions occurring as direct and immediate response to particular stimuli uniquely correlated with them. Two types of reflexes are known: unconditional and conditional. Unconditional reflexes are responses which are stored in the memory of system prior to the operation (stored within the look-up table og the transition and output mappings). Conditional reflexes are responses that are learned as a result of experience. The latter is understood as a chain

R: S$_o$® (S-S$_o$)® E® DM® A

where **So**- what has happened, the observed state, (an ability to *perceive* , *represent*,

[41] which means that under the accepted goal set, the required mappings are to be determined

store the representation temporarily, and *focus* upon the subset of interest is presumed),
$(S\text{-}S_o)$-the change observed, the difference between the previous state and the observed state(an ability to *compare* the representations is presumed),
E-evaluation of the change observed, (an ability to *evaluate* representations is presumed e.g. to assign a measure of "goodness" which is required to extract a decision from the look-up table),
DM-generation of the response, or decision making (an ability is presumed to *generate* the response which means that either a storage of responses is used, or the response can be inferred, or otherwise synthesized).
A-response (an ability to act corresponding to a particular G, i.e. actuation is presumed).

So, the simple act of feedback compensation is in fact, an operation of the response controller which consists of elementary (computational) operations of (1) perceiving, (2) representing, (3) storing, (4) comparing, (5) evaluating, (6) generating the response, and (7) acting correspondingly. It is also implied that the difference **$(S\text{-}S_o)$** is computed over a time interval **Dt** within which the response is not yet being generated. It is also implied that there exist a mapping (a production system, or pre-solved differential equations) that evokes the transforms

$(S\text{-}S_o)$® E® G.

However, a question emerges: why a little action should be determined upon the observation of the little change in a state? Should not the *sequence of states*, the *string of states* be considered rather than a single state? The answer is quite simple: we do consider a sequence of states since a single state at a level of resolution [i-th] under consideration can be expressed as a sequence of states at the higher resolution level [(i+1)-th] with the time discrete **Dt i < Dt i+1**. The chain **R** is applicable to each of the states at the lower level which in turn, is a string of states of the lower level, and so on. So, at a level of resolution the reflexive chain is valid in the above form.

Therefore, it is clear that the chain **R** presumes one more operation (the 8-th one): generalizing time strings at the lower level into single states at the upper level. Certainly, this consecutive refinement can be continued top down that leads to a *nested hierarchical controller* [67]. It is possible to show that familiar procedures of recursive estimation are based on the reflexive chain **R**, too.

Interestingly enough, we cannot include this operation to the set of elementary operations listed above since it is a special operation: it is a part of all other seven operations of the list. Indeed, the first operation (perception) is based upon grouping the units of sensory information into entities, these entities into entities of even lower resolution, and so on. This recursive grouping of units with subsequent formation of entities is also a generalization (not in temporal domain but in the spatial one). Operations of representing, and storing are utilizing the operation of generalization absolutely in the same way as it is shown for perception.[42] The operations of evaluation and of comparison do not incorporate

[42] Obviously, more detailed treatment of these processes (perception, representation, storing) can illustrate the universal character of the generalization process in a more graphic way.

generalization per se, but rather are generalization dependent.[43] (Generalization is focused upon in [42]).

The reflexive chain is described as if it is applied on line. One can easily imagine that similar chain can be activated off-line which brings us to preplanned program of operation. This program is generated by a subsystem of planning ("off-line control generator" in Figure 2). This subsystem reproduces the *"reflexive plan-chain"* before the actual process started:

$$\mathbf{R_p{:}S_e\text{®}\,(S\text{-}S_e)\text{®}\ E_n\text{®}\ DM_n\text{®}\ P}$$

where **S_e**- what expected to happen, the expected state,
(S-S_e)-the change expected, the difference between the previous state and the expected state,
E_n-evaluation of the expected change observed, planning evaluation,
DM_n-generation of the response which is performed off-line, (planning decision making),
P-planned response (a plan of action corresponding to a particular G when the expected change takes place).

One can see that **E_p, DM_p,** and **P** do not require any changes for the rest of preplanned process of motion. In fact other operators can be used since the operation is being done off-line with no computational time limitations, therefore more precise and thorough computational procedures can be applied. It is also easy to understand that planning reflects the so called, *open loop control*.

Here we can see that planning presumes off-line learning in the sense that information about expected circumstances is stored and analyzed prior to operation, a variety of expected situations are analyzed, and the recommended solutions for these situations are *learned* in advance (off-line). Statistically speaking, plan is supposed to represent the (time dependent) statistical expectation of the motion to be executed, or in other words, plan is an expectation of the non-stationary stochastic (vector) process[44]. Any deviations of the reality from the preplanned process are taken care of by the on-line reflexive feedback of the so called *closed loop control*.

Learning Control System is a development of the Control System that includes one subsystem more: the **LEARNER** which functions are:

(1) to enhance the vocabulary[45] of *perception*, i.e. to get a capability to perceive more entities that were available for perception at the beginning of its operation,

(2) to enhance the vocabulary of *representation*, i.e. to get a capability to represent more entities that were available for representation at the beginning of its operation,

[43] These operations can be also described in more detail.

[44] It is well known that there is no theory for finding expectations of non-stationary stochastic processes. Yet, there is a multiplicity of applied methodologies for dealing with this issue.

[45] We can understand now that enhancement of vocabulary can be considered a recursive procedure and when initiated at one of the resolution levels it can trigger an avalanche of corresponding changes at the rest of the levels.

(3) to enhance the vocabulary of storage, i.e. to get a capability to *store* more entities that were available for storing at the beginning of its operation,

(4) to enhance the vocabulary and the arsenal of rules for *comparison*, i.e. to get a capability to compare more entities that have been available at the beginning of the operation,

(5) to enhance the vocabulary, the arsenal of rules, and the precision of *evaluation*, i.e. to get a capability to evaluate more entities that were available, and taking in account more factors than it was in the beginning of the operation,

(6) to enhance the vocabulary, the arsenal of rules, and the set of meta-rules of the *response generating* production system, i.e. to get a capability to generate more adequate responses that were available in the beginning of the operation,

(7) to enhance the vocabulary of responses available for *actuation*,

(8) to enhance the number and the adequacy of the techniques for generalization.

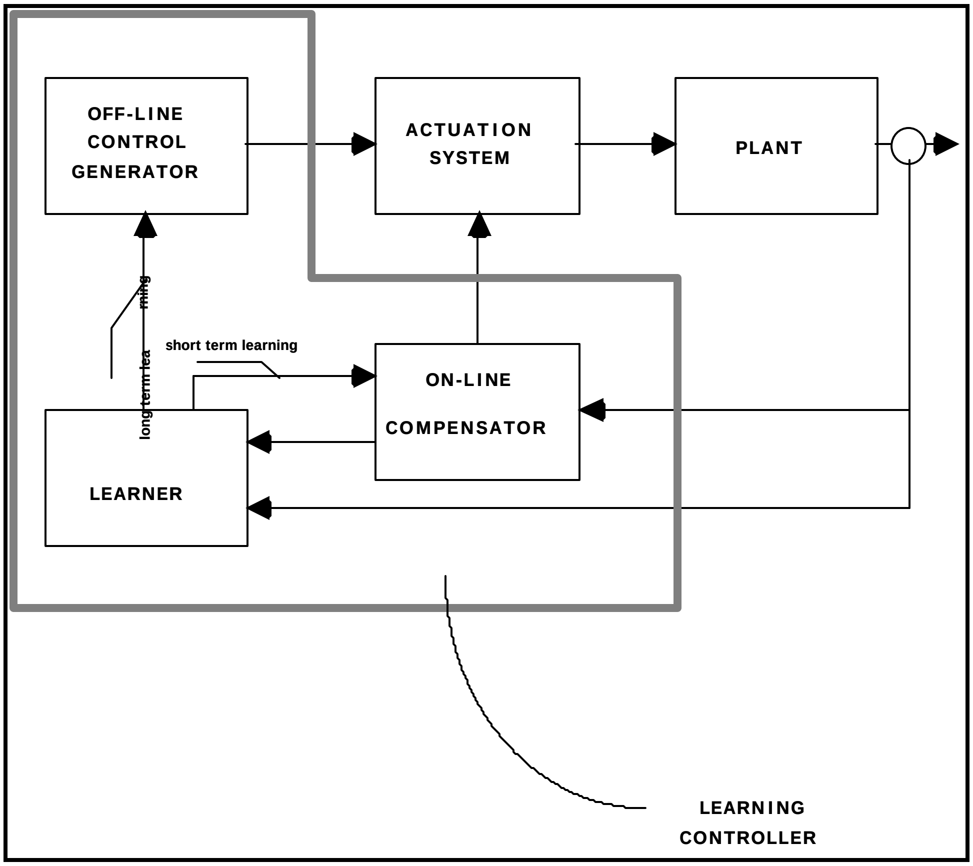


Figure 10-3. Learning Control System

Learning Control Systems introduced axiomatically are consistent with the following definition by G. Saridis [70, p. 22]: "The system is called learning if the information pertaining to the unknown features of the process or its environment is learned, and the obtained experience is used for future estimation, classification, decision, or control such that the performance of the system will be improved". (The meaning of the term learned should be interpreted according to the Definition 11 from **Section 10.1**).

It would be instrumental to consider also the following definition containing a condensed representation of the processes characteristic for Learning Control Systems.

Definition 10. *Functioning of Learning Control System*
Functioning of Learning Control Systems consists of the system of nested hierarchical generalizations performed over the redundant stored information about the current and/or prior experience of operation. This system of generalizations changes the world representation as well as the algorithms of control available for selection during the problem solving.

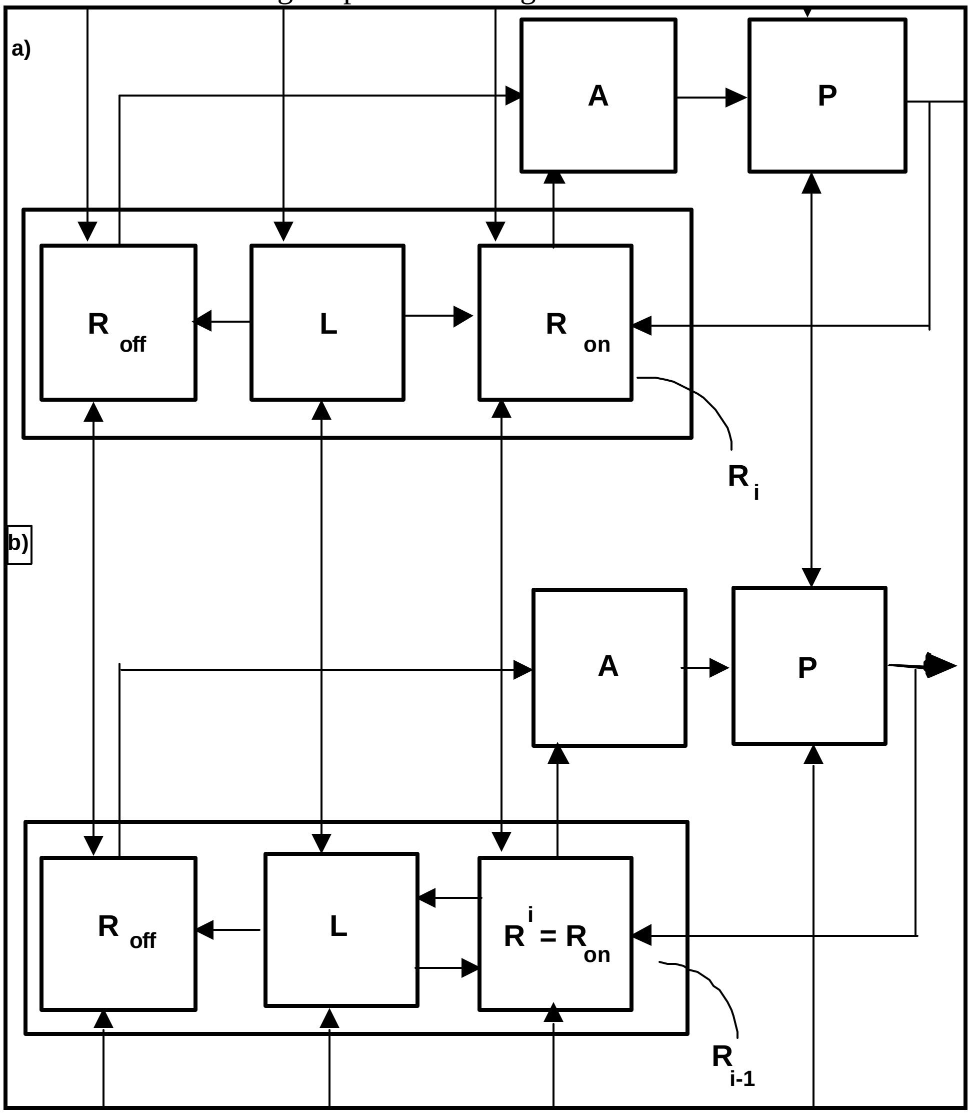


Figure 10-4. Nested hierarchy of learning control system

Figure 10-3 has been redrawn in a simplified form (Figure 10-4,a). Then, the three blocks of the elementary learning controller (R_{off}, L, R_{on}) can be substituted by a single R_{0n} controller for the upper level of resolution (due to the fact that the time-sequence of states at the i-th level correspond to a single generalized state at the (i-1)-th level of resolution. However, a new Learner, and a new R_{off} subsystems are required. This process can be continued bottom-up until the required resolution of planning is achieved.

10.5.3 Emerging Non-conventional Issues

Given both definitions: of the classical approach to control problems, and of the Learning Control System, the following issues can be raised.

a. At the present time, a model is implicitly considered to be the source of the theories for problem solving, not vice versa. The body of conventional control requires having a model of a system as a starting point for the process of design, and then proposes methods of dealing with this formal body of a particular problem within the boundaries of the model selected. The user is more interested in the theory of control that won't depend on the model of system. It is a fact that the **required mapping is capable of inducing the process of interest**, and **we must drive this process toward the goal**. As a result, the deficiencies of the off-line assume model might affect the structure of learning system and learning processes. The models formulated from the general premises do not satisfy the conditions of on-line learning process: they have been created for the off-line use.

b. The problem of constructing the required mapping capable of inducing the process under consideration is becoming a central problem of the research in the area of learning control systems. Indeed, if the model is not supposed to induce the methods of dealing with the system, another way of constructing can be formulated as follows:

-The goal set is formulated by the user. This induces the initial descriptive structure for the subsequent solution**.**

-The implementation set is selected with the participation of the user. These sets of information (the goal-set and the implementation set) supplement the initial descriptive structure. The new descriptive structure can be called the **pre-learning knowledge structure**.

-It would be prudent to use this pre-learning knowledge structure as the starting point for the process of finding the initial control mappings.

-The initial control mappings should be exercised unless they contradict some external knowledge of the designer[46].

[46] A question may be raised: why not to submit this knowledge to the knowledge base of the future machine by supplementing the initial knowledge set before making any trial. The answer is simple: the designer should make a mental experiment, and if the result of it shows that this knowledge is required, then of course, the initial knowledge base must be enhanced.

> -The first results of the operation will contain information which should be considered a source for a (hopefully) converging process of enhancing the descriptive structure, and thus **improving the control mappings**[47]. We would presume that improvement of control mapping is a primary issue of the problem, as opposed to the issue of "adequate descriptive structure".

It is important to realize that the strategy proposed is not a strategy *to learn how to learn*. It is just an evolution of the existing methods of consecutive approximation, and recursive estimation to the less uniform domain.

c. Usually the formulation of control problem is not reconsidered during the control operation. Thus, the goal subsets, and their negotiation is not considered to be a part of control problem. Dealing with the "natural" cost functions is considered to be outside of the functioning of the control system. Eventually, the reality of existing cost-functions is usually not negotiated as a part of functioning of the controller.

d. Problems of dealing with information (knowledge?) are not considered to be part of functioning of the existing control systems. It was a substantial innovative move when K-S. Fu started openly talking about the control systems with *recognition in the loop* [73-75]. Processes of recognition[48] can affect the control processes drastically since the vocabulary of the controller may depend on the recognition results. Learning controllers are expected to work with a constantly changing vocabulary that is expected to be updated as a matter of its normal operation. Each change in the vocabulary may result in the avalanche of generalizations which can totally change the structure of information and the result of control computations.

e. The initial problem formulation and assignment of the goal-subset, is also a part of the control problem for Learning Control Systems

Finally, it is a rule, not an exception that initially the task is posed imprecisely, that the description of many functions to be provided by a computer control system, is incomplete, at least imprecise, and the expected situations of the operation are given to a control engineer by the user in an approximate way. In many cases it is determined by the very fact of impossibility to transfer the knowledge of a user, to the designer . In many cases it is determined by intractability of the problem as it seems to the user. The control engineer is designing the substantial part of the future system (starting with task formulation and ending with actuators operation).

However, often the designer is not familiar with the context of operation. Long time ago, it was understood that the design of the control process, is a trade-off with the parameters of the system. Later it became clear, that it is a trade-off with the model as well as the actual structure of the system. It is becoming clear now that the trade-off includes the processes of task formulation. Moreover, the negotiation of the task-set is the part of design. We can expect that in the future systems, the continuous process of task negotiation will be a part of the computer control system .

10.5.4 Exosystem and Multi-actuation

[47] and here the question will arise: what is the general method of improving them.

[48] Recognition of what?

Information about exosystem (**EXO)** often cannot be given in the beginning, this information should be recognized within the reality of **EXO,** and after recognition, it should be dealt with properly. This information contains facts of different importance (resolution?), and they should be processed in a different way: some of them should be focused the attention upon, some of them should not, and they can be considered in a generalized manner as a part of more general entity unifying a variety of facts. Thus, the idea emerges of studying the reality of **EXO** under different resolutions depending on our interest in the details. Surely, the consistency is to be provided, for the overall system using the different parts of **EXO** considered under different resolutions.

The easiest way of providing this consistency, is to consider the **EXO** as a hierarchy of decomposition in parts, or multiresolutional hierarchy which happens to be the same (this concept coincides with the *frame* concept in AI). This is how the perceptual subsystem delivers the EXO to the controller. The hierarchy of **EXO** appears in a way which is quite similar to the way in which the hierarchy of **PL** appears: the bulk of the information (knowledge?) about the world, or the *world description* should be decomposed (tessellated) into a hierarchical multiresolutional set of parts. The mechanism of the new knowledge acquisition is provided by a system of multiple sensors, their signals must be integrated in order to enrich our knowledge of **EXO**, in other words, **EXO** should be constantly *perceived*. Perception of **EXO** should be considered a part of normal system operation, thus learning is a natural procedure which requires recognition of familiar features and objects as well as discovery of unfamiliar entities and concepts.

On the other hand, since most of the systems are *multi-actuator* systems [79], their actuators must be assigned individual controllers, and each of them should be dealt with as an individual control system within its *scope of attention*. At the same time, all this multi-actuator system can be considered as an object of control: a single system is supposed to coordinate the activities of the multiple controllers subsystems. A separate controller is supposed to submit to the *coordinator* a control assignment which is obtained at a higher level of the control hierarchy (lower level of *resolution*) and is dealing with the objects, parameters, variables, and controls formulated at a higher level of *generality*.

Thus, we came to the hierarchically intelligent controller which has organization, coordination, and hardware control levels as described in by G. Saridis [77, 78]. The *organization level* accepts and interprets the input commands and related feedback from the system, defines the tasks to be executed, and segments it into subtasks in their appropriate order of execution [80]. At the organization level appropriate *translation and decision schemata* linguistically implement the desirable functions [81, 82].The coordination level receives instructions from the organizer and feedback information for each subtask to be executed and coordinates execution at the lowest level. The lowest level control process usually involves the execution of a certain motion and requires besides the knowledge of the mathematical model of the process, the assignment of end conditions and a performance criterion (cost function) defined by the coordinator [83].

References to Section 10.1 through 10.5

[1]. *New Webster's Dictionary of the English Language*, Delair Publ., 1985

[2]. *Encyclopedia Britannica. Micropedia.* Vol.VI, 1978

[3]. G. A. Kimble, *Hilgard and Marquis' Conditioning and Learning*, 1961

[4]. A. Newell, J. C. Shaw, H. A. Simon, "A Variety of Intelligent Learning in a General Problem Solver", Eds M.C. Yovits, S. Cameron, *Self-Organizing Systems*, Proc. of an Interdisc. Conf., May 1959, Vol.2, New York: Pergamon Press,1960

[5]. H. A. Simon, "Why Should Machines Learn", Eds. R. S. Michalski, J. G. Carbonell, T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, Vol. I, Palo Alto, CA: Tioga Publ., 1983

[6]. M. Minsky, *The Society of Mind*, New York: Simon and Schuster, 1985

[7]. R. S. Michalski, "Understanding the Nature of Learning: Issues and Research Directions", Eds. R. S. Michalski, J. G. Carbonell, T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, Vol. II, Los Altos, CA: Morgan Kaufmann Publ., 1986

[8.] J. P. Guilford, "Factorial Angles to Psychology", *Psychological Review*, Vol. 69, 1961, p.p. 1-20

[9]. M. I. Shlesinger, Learning of Image Recognition, *Encyclopedia of Cybernetics*, Vol.2, Kiev, 1975 ,

[10]. Y. Tsypkin, *Adaptation and Learning in Automatic Control*, New York: Academic Press, 1971

[11]. P. S. Churchland, “Neurophilosophy: Toward a Unified Science of the Mind/Brain”, MIT, 1986
[12] G. M. Edelman, The Remembered Past: A Biological Theory of Consciousness, Basic Books, New York, 1989

[13] B. Russell, *The Problems of Phylosophy*,

[14] D. F. Bowers, *Atomism, Empiricism, Scepticism*, Princeton University: Princeton, 1941

[15] K. Koffka, *Principles of Gestalt Psychology*, New York: Hartcourt Brace, 1935

[16] A. Neel, Theories of Psychology: A Handbook, Cambridge, MA: Schenkman Publ., p.p. 319-332, 1977

[17] C. Ehrenfels, On Gestalt-Qualities, *Psychological Review*, Vol. 44, p.p. 521-524, 1937

[18] W. S. Sahakian, *History and Systems of Psychology*, New York: Wiley, p.p. 189-193, 1975

[19] R. Lowry, *The Evolution of Psychological Theory*, Chicago: Aldine Atherton, p.p. 205-208, 1971

[20] D. McCarthy, "Gestalt as Learning Theory", Ed. by G. I. Brown, *The Live Classroom: Innovation Through Confluent Education and Gestalt*, New York: Viking Press, p.p. 46-51, 1975

[21] G. H. Bower, E. R. Hilgard, *Theories of Learning*, Englewood Cliffs, NJ: Prentice Hall, p.p. 49-57, 1981

[22] R. M. Cagne, *The Conditions of Learning*, New York: Holt, Rinehart and Winston, 1965

[23] J. L. Brown, *The Evolution of Behavior*, New York: W.W. Notrton and Company, 1975

[24] C. L. Hull, *Principles of Behavior*, 1943; survey of the results is given in [21].

[25] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, New York: Wiley

[26] W. S. Jevons, *The Principles of Science: A Treatise on Logic and Scientific Method, (1873)*, New York: Dover Publ., p.p. 218-269, 1958

[27] A. N. Kolmogorov, *Foundations of the Theory of Probability, (1933)*, New York: Chelsea Publ., 1950

[28] J. H. Holland, K. J. Holyoak, R. E. Nisbet, P. R. Thagard, *Induction: Processes of Inference, Learning, and Discovery*, Cambridge, MA: MIT Press, 1986

[29] K. R. Popper, *The Logic of Scientific Discovery (1959)*, New York: Harper Torchbooks, 1968

[30] L. Valiant, “A Theory of the Learnable”, Comm. ACM, v. 27, No. 11, 1984, pp. Ii34-1142

[31] M. Vidyasagar, A Theory of Learning and Generalization, Springer-Verlag, London, 1997

[32] T. Mitchell, Version Space, PhD Dissertation, Stanford University, 1978

[33 R. Michalski, et al, “The Multi-purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains”, In Proceedings of National Conference on AI, MIT Press, 1986

[34] R. Quinlan, “Learning Efficient Classification Procedures and Their Application to Chess Endgames”, In Machine Learning, ed. R. Michalski, J. Carbonell, T. Mitchell, Morgan Kaufmann, 1983

[35] S. Vere, “Multilevel Counterfactuals for Generalizations of Relational Concepts and Productions”, Artificial Intelligence, v. 14, pp. 139-164

[37] A. Meystel, J. Eilbert, L. Venetsky, S. Zietz , "Baby-robot: Learning to control goal oriented behavior," Proc. IEEE Workshop on Intelligent Control, pp. 182-186, 1985

[38] "Baby-robot: On the analysis of cognitive controllers for robotics," Proc. IEEE Int'l Conf. on Man & Cybernetics, pp. 327-222, Tuscon, AZ, Nov. 11-15 1986

[39] D. Angluin, “Learning Regular Sets from Queries and Counterexamples”, Information and Computation, V. 75, No. 2, 1987

[40] A. Lacaze, M. Meystel, A. Meystel, “Multiresolutional Schemata for Unsupervised Learning of Autonomous Robots for 3D Space Application”, Proc. of the 1994 Goddard Conference on Space Applications of AI, Greenbelt, MD, 1993, pp. 103-112,

[41] J. Albus, A. Lacaze, A. Meystel, “Algorithm of Nested Clustering for Unsupervised Learning”, Proc. of the 10th IEEE Int’l Symposium on Intelligent Control, Monterey, CA 1995

[42] J. Albus, A. Lacaze, A. Meystel, “Multiresolutional Intelligent Controller for Baby-Robot”, Proc. of the 10th IEEE Int’l Symposium on Intelligent Control, Monterey, CA 1995

[43] C. Watkins, Learning from Delayed Rewards, PhD Dissertation, Cambridge University, 1989

[44] W.-M. Shen, Autonomous Learning from the Environment, Computer Science Press, New York, 1994

[45] Eds. R. S. Michalski, J. G. Carbonell, T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, Vol. I, Palo Alto, CA: Tioga Publ., 1983

[46] Eds. R. S. Michalski, J. G. Carbonell, T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, Vol. II, Los Altos, CA: Morgan Kaufmann Publ., 1986

[47] W. R. Ashby, *Introduction to Cybernetics*, London: Chapman and Hall, Ltd,1956

[48] Eds. C. G. Davis, S. Jajodia, P. A. Ng, R. T. Yeh, *Entity-Relationship Approach to Software Engineering*, Amsterdam : North- Nolland, 1983

[49] Eds. H. Gallaire, J. Minker, J. M. Nicolas, *Advances in Data Base Theory*, Vol.1, New York: Plenum Press, 1981

[50] N. V. Findler, *Associative Networks: Representation and Use of Knowledge by Computer*, New York: Academic Press, 1979

[51] Eds. M. L. Brodie, J. Mylopoulos, J. W. Schmidt, *On Conceptual Modelling*, New York: Springer Verlag, 1984

[52] M. Aizerman, E. Braverman, and L. Rozonoer, "Theory of the method of potential functions for the problem of automatic classification of input functions", *Automation and Remote Control*, v.25, No.6, 1964

[53] M. Aizerman, E. Braverman, and L. Rozonoer, "Probabilistic problem of the automata learning to recognize classes and the potential functions method", *Automation and Remote Control*, v.25, No.9, 1964

[54] M. Aizerman, E. Braverman, and L. Rozonoer, "Method of potential functions applied for the problem of restoration of the functional characteristics of a transducer using random observations", *Automation and Remote Control*, v.25, No.12, 1964

[55] P. H. Winston, "Learning by Augmenting Rules and Accumulating Censors", Eds. R. S. Michalski, J. G. Carbonell, T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, Vol. II, Los Altos, CA: Morgan Kaufmann Publ., 1986

[56] A. Amarel, "An approach to automated theory formation", Eds. H. Von Foerster, G. W. Zopf, Jr., *Principles of Self-Organization*, Pergamon Press, 1962

[57] A. Meystel, "Theoretical Foundations of Planning and Navigation", *International Journal of Intelligent Systems*, 1986

[58] I. Rosenfeld, *The Invention of Memory: A New View of the Brain*, New York: Basic Books, 1988

[59] S. Freud, "The The Interpretation of Dreams", in The Standard Edition of the Complete Psychological Works of Sigmund Freud, Vol. 5, p. 538, London: The Hogart Press and The Institute of Psychoanalysis, 1959

[60] S. Freud, Letter 52, p.p. 233-235, in *The Standard Edition of the Complete Psychological Works of Sigmund Freud*, Vol. 5, p. 538, London: The Hogart Press and The Institute of Psychoanalysis , 1959

[61] G. M. Edelman, "Group Selection as the Basis for Higher Brain Function", Ed. F. O. Schmitt, et al., *The Organization of the Cerebral Cortex*, MIT Press, 1981

[62] G. M. Edelman, L. H. Finkel, "Neuronal Group Selection in the Cerebral Cortex", Eds. G. M. Edelman, W. E. Gall, W. M. Cowan, *Dinamic Aspects of Neocortical Function*, Wiley, p.p. 653-695, 1984

[63] G. M. Edelman, G. N. Reeke, Jr., "Selective Networks and Recognition Automata", *Annals of the New York Academy of Sciences*, Vol. 426, p.p. 181-201, 1984

[64] G. M. Edelman, *Neural Darwinism: The Theory of Neuronal Group Selection*, New York: Basic Books, 1987

[65] P. A. Ligomenides, "Real-Time Capture of Experiential Knowledge", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 18, No. 4, July/August, 1988

[66] A. A. Giordano, F. M. Hsu, *Least Square Estimation with Applications to Digital Signal Processing*, New York:Wiley, 1985

[67]. A. Meystel, "Intelligent Control in Robotics", *Journal of Robotic Systems*, Vol. 5, No. 4, 1988

[68]. P. L. Falb, *Direct Methods in Optimal Control*, New York: McGraw Hill,1969

[69]. J. D'Azzo, C. Houpis, *Linear Control Systems: Analysis and Design*, New York: McGraw Hill, 1981

[70]. G.N. Saridis, *Self-Organizing Control of Stochastic systems*, Marcel Dekker, Inc., 1977.

[71]. R. E. Kalman, P.L. Falb, M. A. Arbib, *Topics on Mathematical Systems Theory*, New York: McGraw Hill, 1969

[72]. L. A. Zadeh, C. A. Desoer, *Linear Systems Theory*, New York: McGraw Hill, 1963

[73]. K.S. Fu, "Learning Control Systems", in *Advances in Information System Science*, Ed. by J.T. Tou, New York: Plenum Press, 1969.

[74]. R.S. Fu, "Learning Control Systems - Review and Outlook", *IEEE Trans. on Automatic Control*, Vol. AC-15, 1970.

[75]. K.S. Fu, "Learning Control Systems and Intelligent Control Systems : An Intersection of Artificial Intelligence and Automatic Control", *IEEE Trans. on Automatic Control*, Vol. AC-16, 1971.

[76]. G.N. Saridis, "Intelligent Robotic Control", *IEEE Transactions on Autonmatic Control*, Vol. AC-28, No. 5, May 1983.

[77]. G.N. Saridis, "Toward the Realization of Intelligent Controls", *Proceedings of the IEE*, Vol. 67, No. 8, August, 1979.

[78]. A. Meystel, "Intelligent Control of a Multiactuator Systems", *IFAC Information Control Problems in Manufacturing Technology 1982*, Ed. by D.E. Hardt, Oxford and New York: Pergamon Press, 1983.

[79]. G. N. Saridis, "Intelligent Control for Robotics and Advanced Automation", in *Advances in Automation and Robotics*, Vol.1, JAI Press Inc., 1985

[80]. J. H. Graham, G. N. Saridis, "Linguistic Design Structures for Hierarchical Systems", *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-12(3), 1982

[81]. G. N. Saridis, J. Graham, G. Lee, "An Integrated Syntactic Approach, and Suboptimal Control for Manipulators and Prostheses", *Proceedings of 18-th CDC*, Ft. Lauderdale, Fl, 1979

[82]. G.N. Saridis, C-S. G. Lee, "An Approximation Theory of Optimal Control for Trainable Manipulators", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-9, 1979

[83]. G. N. Saridis, "An Integrated Theory of Intelligent Machines by Expressing the Control Performance as Entropy", *Control-Theory and Advanced Technology*, MITA-Press, Tokyo, Japan, Vol. 1, No. 2, 1985

[84]. A. Meystel, "Nested Hierarchical Intelligent Module for Automatic Generation of Control Strategies", In *Languages for Sensor-Based Control in Robotics*, ed. by U. Rembold, and K. Hormann, NATO ASI Series, Vol. 29, Berlin: Springer-Verlag, 1987

[85] D. 0. Hebb, *The organization of behavior*. New York:Wiley. 1949

[86] B. Mirkin, Kluwer Academic, 1995

[87] S. Mitra, S. Pal, “Self-Organizing Neural Network as a Fuzzy Classifier”, *IEEE Trans. On SMC*, Vol. 24, No. 3, 1994, pp. 385-398

[88] J. C. Spall, “Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation”, *IEEE Transactions on Automatic Control*, Vol. AC-17, No. 3, March 1992, pp. 332-341

[89] M. Kubat, I. Bratko, R. Michalski, “A Review of Machine Learning Methods”, Machine Learning and Data Mining: Methods and Applications, Ed. by R. Michalski, I. Bratko, M. Kubat, Wiley, 1996, pp. 3-71

10.6 Learning and Behavior Generation: Constructing and Using MR Representation and Goal Hierarchies

This Section introduces an approach for analysis of learning systems as a part of processes for generating behavior. This approach seem to be appropriate for living and artificial intelligent systems. We will introduce an automaton with joint learning and behavior generation which determines a class of learning algorithms for unsupervised learning. These algorithm employ recursive processes of grouping, focusing attention and combinatorial search that lead to the top-down and bottom-up computations with growing multiresolutional hierarchy of knowledge representation. As a part of Behavior Generation, Learning searches for an appropriate set of rules (control law K), and thus, for a preferable motion trajectory. These planning/control processes are based on and include further development of multiresolutional knowledge representation and lead to a temporal evolution of the system. Analysis of this automaton demonstrates benefits of a joint analysis of spatial and temporal processes of behavior and learning.

10.6.1 Learning from Multiple Experiences

In 1986 a concept of ``baby-robot" was introduced in [1]. This concept emphasizes processes of unsupervised learning and is associated with the joint evolution of behavior and knowledge incorporated within a system. It seems plausible that learning from multiple experiences is inseparable from the architecture of applying the acquired knowledge for shaping the desirable behavior. We will not focus upon analysis of the evolution of living creatures, or problems of knowledge acquisition and growth of the intelligence that is a result of the evolution. All survival oriented matters are beyond the discussion in this Section. Instead of analyzing living creatures, we introduce a formal system - a learning automaton that can evolve as a result of its own experiences. It has the faculties required to obtain and use knowledge: sensors, subsystems for storing and organizing information, subsystems for generating commands. It has a goal that is prescribed externally. It also has faculties to produce its own behavior: actuators for changing the world. However, initially it has neither any world model nor any rules to achieve its goal. This knowledge should be learned, and we would be interested in understanding how does it happen.

The concept of experiences that emerge as a result of behavior is linked with the idea of "goodness". The automaton should be capable of evaluating the desirability of its own behavior and parts of this behavior: its components –elementary experiences as discrete events (this will be introduced and explored in sub-section 10.4.3). This will allow to judge how beneficial are the processes of this automaton "thinking", since the output behavior is shaped by the actions that emerge as a result of plans: strings of decisions how to achieve goal-states, what are the required actions leading to goal-states, and how they should be organized in strings. This process of finding the string of desirable states and actions leading to them was introduced in Chapter 8 as *planning*. Planning would be impossible without an ability to judge the degree of "goodness": the desirability of states and actions would be acquired in advance. *Execution* as the process of generating and applying proper commands to execute the planned trajectory and compensate for errors (see Chapter 9) is the generator of "experiences". Both planning and execution are a part of control system functioning. The process of acquiring the relevant information and processing it so that a proper behavior could be generated is called learning. Planning/control and learning are complementary computational procedures, merging into a joint process of intelligent control.

The process of planning starts with focusing attention upon a subset of representation: the automaton should select the "subset of interest": the initial representation of the world that will be explored, or the sub-space with its boundaries. The latter will be fragmented ("discretized") into tessellata which determine the resolution of maps', or their granularity (or scale). Combinatorial search is performed as a procedure of constructing all possible strings of consecutive strings of states and choosing one of them (the minimum cost string). Focusing attention is instrumental in limiting the multiplicity of all possible strings formed out of the space tessellata at a particular level of resolution.

Grouping the tessellata in a variety of feasible strings and the subsequent selection of the "best" string allows for focusing attention upon the vicinity of the chosen string. This initiates the process of "zooming-in" and leads to the evolution of the behavior by consecutive refinement of the planning process top-down. The latter is performed by limiting the scope via constructing an envelope for planning at higher resolution around the minimum cost string. This envelope is being submitted to the next level of resolution where the next cycle of computation starts. Focusing attention presumes proper distribution of nodes in the state space so that no unnecessary search be performed. This is how combinatorial search forms the alternatives at all levels of resolution. Grouping, focusing attention and combinatorial search together can be considered *generalization* which in the case of our learning automaton is a process of generating plans at a particular level of resolution and generation maps for the subsequent planning at the higher level of resolution. The intelligent properties of the learning automaton are produced by joint functioning of these three operations: grouping, focusing attention and combinatorial search (GFACS). It will allow the automaton to ``behave'' as a result of its planning decisions and actions of the actuators, create experiences, evaluate them and learn from them. The overall automaton[49] can be represented using the concept of elementary loop of functioning as shown in Chapter 2 [2-4].

A novel mechanism of unsupervised learning, with combinatorial enhancement and generalization, is introduced in this sub-section and explored in the subsequent ones. It is demonstrated that this learning leads to emergence of a multiresolutional representation, a hierarchy of subgoals, and a hierarchy of control commands. This type of learning system provides an explanation of evolutionary processes both in technical objects and in living creatures. The evolution of knowledge is discussed. We will imply that the evolution of living creatures is a special case of the evolution of knowledge representation and their behavior generation.

The process of evolution of the learning automaton leads toward and can be characterized by a gradual sophistication of its knowledge and behavior. This gradualness is not monotonic in all its derivatives. The evolution of knowledge is rather ``punctuated[50]''. It progresses in steps that emerge when a collection of hypotheses gets transformed into a rule. The evolution of behavior has similar steps due to the emergence of new skills which happens also in different moments in time. We intend to use our model for processes both of early and mature learning. In final sub-sections, a unified process of learning and behavior generation is discussed.

10.4.2 Algorithms of Unsupervised Learning

Learning Automaton. An automaton is a state machine which generates outputs by using its transition function and state-output function tabulated in advance for all possible situations.

[49] “Learning Automaton” introduced in this Section and based upon the concept of ELF and GFACS subsumes the concept of “learning agent” known in the domain of AI and consisting of “problem generator”, “performance element”, “critic” and “learning element”.

[50] See Lewontin, “Theory of Evolution”

Automata are capable of demonstrating ``reactive behaviors" according to the prescriptions stored in their transition and output functions. A class of learning automata is introduced which are state machines whose transition and state-output functions are updated and modified based upon prior experiences which are stored in the memory and transformed into sets of rule. Thus, a meta-control structure is presumed which modifies the transition and output functions.

The transition and output functions of the learning automata have open lists of rules. New rules can be added to these lists. The aim of a learning automaton is to acquire rules that will increase the total reward in achieving the goal, or to increase the value of an objective criterion of optimality associated with the goal achievement. This concept is similar to the one described in [5]. In this Section, we will equip learning automaton with a new capability: to synthesize their output by combining together previously stored rules in search of the most appropriate behavior. Thus, in addition to reactive responses, our learning automata demonstrate the skill of deliberation, or planning.

This new type of automaton will be called Learning and Planning Automaton (LPA). LPA is presumed to have a learning system LS which allows for enriching both the transition and the output functions and a planning system which is a part of the mechanism of behavior generation BG. We will demonstrate that as the system of rules develops it becomes a hierarchical one and creates corresponding developments in the input and output vocabularies of LPA. This is equivalent to formation of the hierarchy of automata as a result of the evolution of a single learning automaton. Operators of LS are equipped by minimum initial set of tools, or ``bootstrap knowledge." This includes the ability to form strings, to construct hypothetical implications, and to infer tautologies.

Learning system LS can be defined as a system of acquisition of experiential information, transformation of the information of experiences into rules of action, derivation of new concepts, and organization of these concepts into knowledge and decision-making structures suitable for achieving the goal.

In order to apply the results of learning, each LPA is equipped by the set of actuators that implement control commands and create the output of learning automaton – Changes in the World. The latter are measured by the Sensors. A set of sensors is the only source of information for LPA about the state of the World (Situation.) The automaton is also equipped by subsystems of Sensory Processing and World Model that allow for interpreting the input from sensors (see ELF in Chapter 2). Each of these modules has been discussed in the previous Chapters. We will treat all modules of LPA as discrete event systems.

Many questions arise at this moment: is this learning automaton based on a Mealy or a Moore automaton? is this a Markov controller? how do we "equip" it with actuators and sensors? is the output of the automaton submitted to the actuators, or the actuators *are* the output of the automaton? is the output of sensors submitted at the input of LPA, or it *is* the input of LPA? what does it mean to equip the automaton with actuators and sensors? does it mean that the World Model emerges within the automaton? how does its transfer function looks like/ what are its states?

The Learning System can judge upon truthfulness of sensor (input) information only by the results of actions (behavior) which are undertaken to achieve the goal. Therefore, LPA should be able to evaluate the results of its behavior. The subsystem of Behavior Generation (BG) contains transition function and output function [2, 3, 6] (see Chapter 7). This section is a further development of the approach presented in [7, 8].

Experiences. We use the term situation to describe the state of the world represented as a set of n sensors outputs which arrive at a moment of time i.

(1) $S_i = \{s_{ji}\}, j = 1,...,n; i=1,...T$

where s_{ji} is a numerical (or logical) value for the j sensor at the time i.

Different levels of resolution will have different sensors, and the representation of the situation can be different depending on the type of representation used (i.e. intervals, NNs, splines, decision trees, etc.). An example using an interval representation could be demonstrated as follows:

$\{ S_{11}1=[10,10.1], s_{21}=[5.6,6.0], s_{31}=[-7.0,-7.0], s_{41}=[15,15.1], s_{51}=[ON,ON]\}$

where s_{11} is the distance to the target at time step 1,

s_{21} is the heading at time step 1

s_{31} is the angle to target at step 1

s_{41} is a laser range distance to the obstacle at step 1

s_{51} is a touch sensor.

and the interval represents the value of the sensor within time step equal to “one”.

If on the other hand, we are representing with NNs, the situation could be:

```
          --------------
$S_{11}$----|          |
$S_{21}$----|          |
$S_{31}$----|    $S_1$    |----- True/false
$S_{41}$----|          |
$S_{51}$----|          |
          --------------
```

where S_1 is an element of a NN that gives True or False by thresholding the its feedforward output when the sensors values are connected to the input.

Action is a set of m action outputs which are generated by the system at moment of time i

(2) $A_{i,i+1}$= {a_{ji}},   j = 1,...,m; i=1,...T

where a_{ji} is a numerical value that has been observed as an output of actuator j.

There are two possible kinds of actions that can be represented in the Learning Automaton.

a) is a set of values that where the output sent to the actuators, for example:

{A_{11}=10, A_{21}=15}]

where A_{11} is the output to the "go forward" actuator at time step 1

A_{21} is the output to the "rotate" actuator at time step 1.

b) is a situation to be achieved, this could also be though of as a subgoal. For example:

{achieve S_3 = [10,11]}

In order to execute this action, a set of actions of the determined type needs to be found to submit it to the actuators.

Situations can be represented as sets (lists), or as vectors within a particular system of coordinates. Actions are understood as causes of changes that are sensed after the actions are applied. This allows for recording correlations between these causes and the sensed changes which introduces the concept of a rule.

We use the term *experience* instead of *cause-effect relation*ship because we want to underscore the fact that no prior cause-effect knowledge is available. Sometimes, the functioning of learning automata is described in terms of the quadruplets: states-before-action, states-after-action, actions, rewards. For example, a ``single move" experience can be described as follows:

(3) $E_{i,i+1} = \{S_i, A_{i,i+1}, S_{i+1}, J_{i,i+1} (G)\}$, i=1,2,...T;

where S_i is a vector of situation at the time i, $A_{i,i+1}$ is a vector of action applied to S_i, S_{i+1} is a vector of the next situation in which $A_{i,i+1}$ has been applied. In dynamic systems, we are interested in longer strings of moves that we regard as experiences. Any concatenation of single moves generates a multi-move experience of the type ``situation-action-situation-...-situation" which serve a basis for the subsequent learning.

$J_{i,i+1}$ (G) is the evaluation of goodness of the action under the goal G. It is the final situation to be achieved as a result of the behavior. The overall goal of functioning is presumed to be given to a system externally. From [2-6] we know that goals can emerge also as a result of planning. From [3,4] we know that the process of learning generates subgoals.

It seams reasonable to evaluate experiences not for just one goal but for as many goals as possible. The tentative experiences during the learning process are obtained as a result of random actions, thus they carry equal amount of rule creating power for each possible goal. By including the goal into the experience set we abandon the universal part of the information obtained.

An example of the experience:

{{ S_{11}=[10,10.1], s_{21}=[5.6,6.0], s_{31}=[-7.0,-7.0], s_{41}=[15,15.1], s_{51}=[ON,ON]}
{A_{11}=10, A_{21}=15}}
{ S_{21}=[10.1,10.7], s_{22}=[5.6,6.0], s_{23}=[-7.0,-7.3], s_{24}=[15.1,16], s_{25}=[ON,ON]}}
$J_{21,11}$(G)=677.5

The description of “situation before action”, “action” and “situation after action” should be supplemented by evaluation of the “goodness” of the result achieved. Evaluation of goodness is a complicated feat: even when the simple distance to the goal is the measure of our accomplishments , we should evaluate goodness of the move by taking in account:

- how much closer LPA to the goal after the move than before
- is there any particular reward associated with the move we performed
- how expensive was this move (the “negative reward”, or “punishment”)
- how much effort it will take to complete the whole assignment after this particular move

- what will be the sum of all future rewards collected during the motion

When the rewards and punishments are evaluated, LPA should apply a general policy of comparative attitude toward the present and future rewards: are the future rewards discounted when taken in consideration “now” and what is the value of discount, and so on.

J(G) is a “goodness” (or “reward “)attained under the goal G as a result of a single move. It is presumed that any valued experience is associated with a certain measure of ``goodness'' $J_{i,i+j}(G)$. It can be interpreted as a reward for the pursuit of the goal G. In a multi-move experience the value of reward is determined not only by the initial and final situations but by the collection of moves that brought LPA from the initial to the final situation. Both single-move and multi-move experiences will be regarded as a unit of experiences. A goal is imposed upon the experiences and the goodness is calculated for each on of them. The value of reward will be used for the subsequent process of hypotheses evaluation, generation, and selection. Ultimately, it determines the rules necessary for ``survival.''

Rules. A rule can be obtained as a result of transforming the cause-effect relations discovered within repetitive experiences. From an experience

$E_{i,i+1} = \{S_i, A_{i,i+1}, S_{i+1}, J_{i,i+1} (G)\}$, i=1,2,...T;

two types of rule hypotheses can be formally deduced with no additional information required:

Type CR: Control Rules:

“IF present situation S_i and goal G THEN action required is $A_{i,i+1}$ “

and

Type ER: Event Rules

“IF present situation S_i and action $A_{i,i+1}$ THEN new situation S_2”.

In the literature, one can find two extreme cases related to these two types of rules:

a) Systems contain only CRs.

Some examples of the behavior that these systems create can be appreciated in multiagent behavior. In this kind of approach, the behavior is created by an almost always fixed set of CRs that determine how the system reacts to the situation. If there are no ERs, it means that no planning can be conducted, since planning requires that many possible states be contemplated.)

b) systems that contain only ERs.

(An example of this kind of behavior is a system employing any path planning algorithms. In the algorithm, an appropriate behavior is found by testing with random points the model dictated by the ERs. In these cases, any error of the model will cause the system to perform replaning, since there are no reactive rules that will give a recommendation of what to do if the required rule is not in the rule-base.)

In this Section, we discuss an LPA where both CRs and ERs are being learned. The ERs are used for finding the next state in the planning algorithm, and the CRs are used to guide the search algorithm towards actions that where more successful in the past.

There are some important differences between ERs and CRs. ERs represent knowledge related to the properties of an environment, and are not related to any particular goal. However, they can be used to create behavior for different goals by testing the model and searching for the appropriate trajectory leading toward the required goal. On the other hand, CRs represent knowledge about the ability of LPA to move within a particular environment toward the particular goal. CRs are valid for a specific goal, and they can be easily used for creating behavior, without testing the environment, only by reproducing what was successful in the past.

Some example of rules are given for illustration.

a) The rules of CR type:

i) using intervals

if S_2 = [4.5,4.8] and S_3=[4.3,4.7] and goal G={S_4=[1.1,1.2]) then A_{23} [10]

which can be interpreted as if the heading and the angle to the target are "equal α and β correspondingly" then DO[A_{23}], or “turn until ‘heading’ is equal to ‘angle-to-target’”.

ii) using an NN

In this case, the same rule could be learned stored for all "S_2=S_3" necessary in the interval representation. Thus, the CR rule is represented as follows:

IF $NNSFF_1(S_1,S_2,S_3,...)$=1 AND GOAL={$NNSFF_2(S_1,S_2,S_3,...)$→ THEN A_{23} [10]

where NNSFF1 is a neural network state in the feed forward mode. Another approach could be to train a separate NN to learn the mapping between the situation and the recomended action:

IF $NNSFF_1(S_1,S_2,S_3,...)=1$ and goal={$NNSFF_2(S_1,S_2,S_3,...)\rightarrow$

THEN NNRFF1(S1,S2,S3,...)

where $NNRFF_2$ is the NN rule mapped in feed forward mode that maps the situation recognized in $NNSFF_1$ to a set of actuator commands.

b) The rules of ER type can be constructed:

a) by using intervals

if S_{2i-1} = [4.5,4.8] and S_{3i}=[4.3,4.7] and A_{1i}=1.1 then S_{2i-1} =[5.1,5.2]

where i is a number of the time step.

b) by using NNs the same ER can be represented:

$NNSFF1(S_{1i-1},S_{2i-1},S_{3i-1},...,S_{1i},S_{2i},S_{3i}, A_{1i}, A_{2i},....)= S_{i+1}$

Certainly, in some cases it may not be possible to train just one NN to represent this mapping. Then, the space can be broken up in smaller parts so that a group of NNs can accurately represent the required mapping.

To find new rules an operation of inductive generalization is applied as follows. Clusters of similar experiences are created. Then, the hypotheses of control rules and event rules are postulated by transforming (inverting) the generalized experiences. These rule hypotheses are used by the automaton as provisional rules for behavior generation. A frequency of valuable rewards estimates the validity of a newly created rules and confirms their correctness if the frequency is high.

After the rules are confirmed, some of their parts that emerged as a result of generalization, may not belong to the initial vocabularies (VI and VO.) These parts are considered to be new concepts. Concept is understood as a label for the entity that is a part of the rule (obtained recursively as a cluster or group of higher resolution inverted experiences). All new concepts are obtained from two available classes of rules (see Figure 10-5). Situations and actions are concepts and so are clusters of situations and actions, as well as their components. They are organized into ``ontology" base that stores them together with their relationships.

The main reason for creating concepts is the need in remembering concept-clusters of higher resolution (in this case, the clusters of sensor readings) and use them to generate future behavior in an efficient manner. These clusters are created only because the system can behave more efficiently if these clusters are constructed. It is prudent to assume that intelligent creatures (e.g. humans) create clusters and assign label to them only because they help to change their behavior in an efficient manner. The latter statement rejects the popular belief that labeled clusters exist NOT within the eye of the beholder.

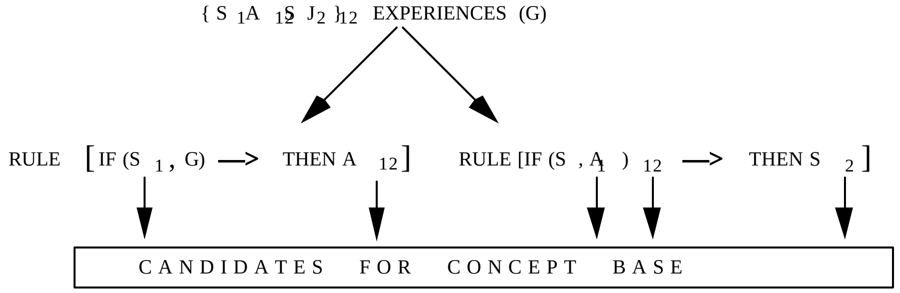

{ S 1A 1S2 J2 }12 EXPERIENCES (G)
RULE [IF (S 1 , G) —> THEN A 12]
RULE [IF (S , A1) 12 —> THEN S 2]
CANDIDATES FOR CONCEPT BASE

Figure 10-5 Experiences -> Rules -> Concepts

If we believe the "clusters for behavior" statement, it is easy to understand that concepts can only be created as a result of interaction between the learning creature and the environment. These interactions are the ones that create new concepts. Obviously, different interactions with environment under different goals can create separate sets of concepts (see Sapir-Wharf hypothesis.) Some experiences are collected by creating random actions by the actuators (no particular goal is stated). Experiences achieve different goodness measure based on the goal at S_1=0. The best of them are selected. Assume, the set of experiences is stored:

S_1	S_2	S_3	S_4	S_5	A_1	A_2
11	35.6	**35.3**	**120**	**off**	0	**10.0**
98	**35.3**	**35.9**	13	**off**	**0.1**	**10.0**
58	**35.8**	35.8	12	**off**	**-0.2**	9.8
235	35.5	35.5	**138**	**off**	0.3	9.9

The values with a tendency to cluster are printed in bold letters. The following hypothesis is extracted from this table:

IF S_2=[35.3, 35.8] and S_3=[35.3,35.9] and S_5 = OFF→

THEN A_1=[-0.2,0.1] and A_2=[9.8. 10.0]

(Note that in S_1 there were no clusters found, and in S_3 the cluster is shifted. The reason for this is that the values in this example cover most of the range. This could be different if the clustering threshold would be different.)

In this case, the triplet {S_2=[35.3, 35.8, S_3=[35.3,35.9], S5 = OFF} and the couple { A_1=[-0.2,0.1], A_2=[9.8. 10.0]} are the candidates for new concepts.

The new concepts, obtained from the generalized experiences as a result of their transformation into rules, are new object-labels and action-labels which are not present in the lists of previously defined by the input and output vocabularies. These novel labels are words of the new vocabularies at the lower, more generalized level of resolution. They describe clustered experiences: the phenomena related to groups of initial units of experiences; these groups are regarded now as new entities belonging to the lower level of resolution. Thus, the single sensor value becomes a group of sensor values that are unified by the fact that a particular action or string of actions lead to highly rewarded results.

Goodness, or reward (J) is computed in a different way depending on a particular application. For example, it can be computed as a difference between increments of cost $C_{i,i+1}$ accumulated for the interval from state i to state (i+1), and $C_{i-1,i}$ accumulated for the interval from state (i-1) to state i.

The following sequence of activities can be explicated from the definition of learning:

1. Experiences {E_i} are collected and stored in memory.
2. Experiences are compared, the values of similarity (resemblance) are determined, and clusters are formed by the virtue of resemblance.
3. Clusters of experiences that already have proven their cause-effect relationships, are transformed into rules, and control rules and events rules are separated.
4. The clusters of situations and actions that are parts of the newly created rules are stored as concepts of lower resolution; then the growth of the concept base begins. New concepts emerge as a result of clustering. These concepts are labeled and receive a status of a new word.
5. The words for the clusters form a vocabulary. This vocabulary is regarded as a lower resolution vocabulary. Thus, in addition to the previous vocabularies VI and VO we obtained two new vocabularies VI and VO .
6. Subsequently, the new experiences are stored in parallel in two new vocabularies: the original one VI and VO , and the one formed by the newly created concepts VI and VO .
7. The process of consecutive operations

{collection of experience→hypothesis formation→generation of rules→concepts emergence}
is repeated each time as a new experience arrives.

8. As vocabularies VI and VO grow, they allow to represent and control functioning of the system by using their new words. Thus, new, generalized experiences can be recorded for generalized situations and actions. The sequence of steps 1-6 can be repeated which will result in building up a new lower level of representation.

This algorithm describes a recursive process which leads to a development of the multiresolutional system of world representation and a multiresolutional systems of rules of actions, both acquired as a result of learning. The described learning process employs only focusing attention, search, and grouping the experiences by their already *existing* features of resemblance. This is not always satisfactory for successful learning. The third component is simplified: no new combinations are formed. Formation of new combinations will be introduced in a form of *combinatorial enhancement*.

Learning with generalization allows for using experiences in a very efficient way. Multiple clustered experiences are treated as group phenomena. Rules generated by them are group rules. When the number of group rules becomes large, the algorithm of generalization can be applied again. The group rules emerge that have even lower resolution.

However, this consecutive generalization is just a tool of reducing complexity. In order to receive innovations, *new combinations* should be introduced for generating words beyond the existing experiences. Similar mechanism has been proven to be very useful for design purposes [10].

Combinatorial Enhancement: Searching for Hidden Implications. Each experience can testify only on behalf of some part of the overall situation. This is why in the theory of learning, the concept of a combinatorial enhancement of rules and concepts has been introduced which produces enhanced actions and situations.

Indeed, the available sensor information not necessarily can be a good basis for generating a hypothesis. Consider an example with autonomous mobile robot that must learn how to act in a particular environment. In a particular situation, a single actuator command cannot be a proper response. Frequently, a combined command (steering + propulsion) must be assigned. Just value of the sensor reading of angle of [``heading'', –a] or [value of ``angle to the goal'', –s] cannot be an antecedent in a rule what to do. However, their difference (–a - –s) is a perfect variable of control.

Such two persuasive experiences of learning as the evolution of living creatures and the evolution of knowledge would be substantially impaired if no combinatorial enhancement would be possible. The following factors are considered critical for stimulating evolution of living creatures: reproduction, mutation, competition, selection [9]. These factors are applied to knowledge in a similar way. Reproduction is like formation of statements of experiences. Competition and selection are like generalization of the best results with the highest values of rewards (see Subsection 2.3.) Search in a form of combinatorial enhancement is the direct analog of mutations. It can be compared to crossover and other mechanisms of forming new entities with an element of randomization.

In this section. we explore only the most primitive combinations: based upon applying arithmetic rules to all possible couples. We call formation of combinations of the available data for the consecutive searching and grouping combinatorial enhancement. For example, if measurable values v_1 and v_2 are known in the beginning of some experience, neither v_1, nor v_2 might be indicative of the need to use a particular action while the value of $(v_2 - v_1)$, or $(v_2 + v_1)$, or other combinations might be important to interpret results of measurements and construct meaningful clusters.

The enhancing takes as input a set of values and gives as output a combination among these values:

$$(4) \qquad \text{Comb } \{ v_1, v_2, v_3,...,v_n \}= \{v_i , (v_i \pm v_j)\}, i=1...n, j=1,...n, i \quad j$$

This operation creates a new set containing the initial set and all possible combinations of the elements of the initial set. In (4) only combinations of additions and subtractions are illustrated. The creation of classes of situation and action introduces two challenges:

a) A suitable language should be found that can accurately represent the classes that the intelligent system is trying to learn. This representation must be compatible with the behavior generation process and should give savings of space (since we cannot store all experiences) and give savings in time of computation (the generator of behavior must use these classes for finding the control)

b) A suitable algorithm should be found that can learn efficiently these classes, and store them in the defined form.

The discovery of classes entails the creation of boundaries that separate the entities in the class from the rest of the set (from other classes). Different approaches have been found for creating these boundaries: from introduction of simple intervals by thresholding or constructing hypercubes to using neural networks. The following heuristic approaches are used to perform these tasks:

1) Creation of intervals (tessellating the space) is probably the simplest way of discovering and storing the classes. There are many methods for finding these intervals. One of the simplest is to look at the situation dimension by dimension, and find places with no good experiences, and use this spaces to separate classes. An extension to this process is the creation of combination of sensor values to ``enhance'' the representation of the experience. For example, by looking at the addition or subtraction of two dimensions, we are in turn allowing the system to be more flexible in finding the shapes of boundaries between the adjacent classes.

Another method of finding intervals is described by Quinlan in C5.0 [22]. The disadvantages are his method is that if the class does not fit the hypercube shape that is imposed by his algorithms, the classes get divided into multiple smaller classes, occupying more space and causing the behavior generator to spend more time to find the control.

2) The closest neighbor algorithm is a very popular method of clustering. This algorithm represents its classes in hyperspheres. The algorithm has several drawbacks: an initial (desired) number of clusters must be specified; classes that do not fit the spherical shape, that the algorithm imposes, wind up divided into many classes. Again, we end up occupying more space than necessary.

3) A tempting approach is the use of Neural Networks (NNs). They can be trained (supervised training) on the complete set of experiences and can be trained so that the output can distinguish between inside and outside of the boundary. The advantage of NNs is that given sufficient amount of neurons they can learn a variety of decision boundaries. The disadvantage is that they may take long time to train in sequential computers, and the results of the learned classification may be hard to interpret (see Section 10.6).

An enhanced representation of a situation is constructed by considering enhanced vectors of situations and actions applied. For example, if the vector of situation was built upon readings of three sensors $S=\{s_1, s_2, s_3\}$, the enhanced vector of situation will include nine coordinates $\{s_1, s_2, s_3, (s_1+s_2), (s_1+s_3), (s_2+s_3), (s_1-s_2), (s_1-s_3), (s_2-s_3)\}$. Similar change will happen to the vector of actions: instead of $A=\{a_1, a_2, a_3\}$ a new vector will emerge: $\{a_1, a_2, a_3, (a_1+a_2), (a_1+a_3), (a_2+a_3), (a_1-a_2), (a_1-a_3), (a_2-a_3)\}$. It would be prudent to consider also a new hybrid vector $\{(a_1+s_1), (a_1+s_2), (a_1+s_3), (a_2+s_1), (a_2+s_2), \ldots , (a_2-s_3),(a_3-s_1), (a_3-s_2), (a_3-s_3)\}$. However, in the simple examples that we used from the area of robotics, it was hard to come up with physical interpretation of each hybrid combination.

Thus, the enhanced situation is a set formed by a situation, all possible combination of its components, and all possible combinations between components of the action A and the situation S. Each experience presented by (3), or a string of single-move experiences (3) should be stored together with the synthesized enhanced situation S_{Enh} and enhanced action A_{Enh}.

The Algorithm of Inductive Generalization. The process of inductive generalization is one of the possible procedures of generalization. In addition to formation of clusters typical for any generalization, inductive generalization uses information about dynamic properties of the statistics of cluster formation. The maxim of inductive generalization claims that multiple occurrences of similar experiences testify for existence of a particular rule, if most of these occurrences have the same (or similar) explanation of causes [14]. If the number of occurrences is not statistically persuasive, then we can talk about the case of hypotheses generation by means of abductive generalization. In both cases, it is important to account for the list of attributes/variables of the situation, and also for the relations among them [15].

The process of unsupervised learning employs the algorithm of generalization which presumes a multiple iteration of the triplet from Figure 10-5: focusing attention on the subset of experiences and searching among them and their combinations until a grouping can be performed, i.e. a cluster of similar units could be substituted by a single generalized hypothesis [10].

This hypothesis is then stored in the database of hypotheses. If the subsequent experiences confirm the hypothesis, it becomes stronger. This is not a trivial operation. Automata with operation of generalization have not been previously discussed in the literature on learning. The operation of generalization includes steps 1-4 of the algorithm of learning.

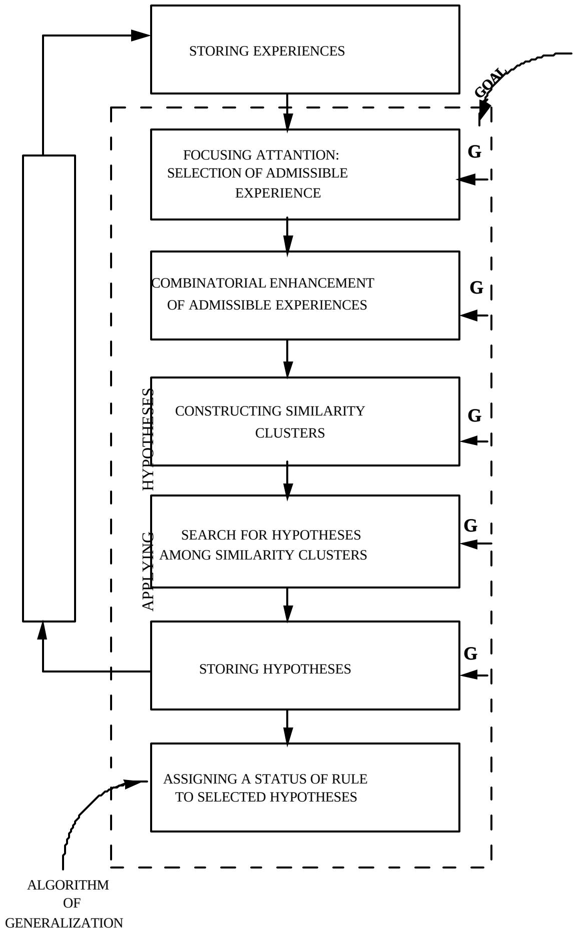

STORING EXPERIENCES
GOAL
FOCUSING ATTANTION: SELECTION OF ADMISSIBLE EXPERIENCE
G
COMBINATORIAL ENHANCEMENT OF ADMISSIBLE EXPERIENCES
G
CONSTRUCTING SIMILARITY CLUSTERS
G
HYPOTHESES
SEARCH FOR HYPOTHESES AMONG SIMILARITY CLUSTERS
G
APPLYING
STORING HYPOTHESES
G
ASSIGNING A STATUS OF RULE TO SELECTED HYPOTHESES
ALGORITHM OF GENERALIZATION

Figure 10-6 The Generalization Algorithm

Thus, the stored experiences and the goal both are inputs to the process of generalization only in the very beginning. Then, the third input to the process of generalization not included into Figure 10-6 is information from the database of hypotheses for behavior generation. This database is useful for the algorithm of generalization, and has two major functions:

- it allows for restoring information that was already lost in erased experiences; the hypotheses included in the database are generalizations (or compressions) of these lost units of information.
- it does not allow for creation of hypotheses which were already created, or creating hypotheses that have been already created earlier but demonstrated to be not valid. Since the algorithm of learning is applied recursively to its own results, each two adjacent levels of resolution use generalization applied to the initial information which can be considered experiences and to generalized information which is

called hypotheses. Thus, we can distinguish two kinds of generalization pertaining to the level to which they are applied: generalization from experiences before any hypothesis is available and generalization from hypotheses.

The following factors are characteristic for the process of generalization and should be taken in account in any concrete computer algorithm. They are applicable both for generalizing experiences and hypotheses. We may treat hypotheses as experiences by themselves. The hypotheses are validated as a result of applying them. Thus, they become subjects for the subsequent generalization which is characterized by the following factors:

1. The most straightforward method of clustering is based upon extraction from the data base of two important subsets, admissible and non-admissible. Admissible experiences are ``good,'' while non-admissible are `bad'' with respect to the given goal which is determined by values of goodness/reward they deliver. The degree of goodness which determines the threshold of clustering can vary. It will determine the productivity of learning and eventually the success of functioning.

2. The admissible set is used to generates hypotheses of rules that prescribe ``what to do'', while the non-admissible set generates hypotheses with a warning content: ``this should not be done'' in this situation.

3. The representation of each experience is enhanced by combining components that can be validated as new concepts. Actually, enhancement plays the role of ``imagination'' in the overall process of generalization. It is similar to creating the strings of anticipated trajectories in the subsystem of Behavior Generation (see Chapter 7).

4. Searching is performed to determine groups of similarity among the enhanced experiences, and the clusters are created.

5. Each of the clusters of similar enhanced experiences is considered to be a candidate for becoming a rule hypothesis.

6. Experiences are tested, and the results of testing enter the base of experiences.

7. Hypotheses are validated by statistics of their use and then enter the base of hypotheses.

All manipulations with sensor/actuator values are performed after the values are normalized. The reason for normalization comes from the fact that different sensor and actuators work over different ranges. Thus, normalization is a function which maps a value from a particular interval into the normalized value interval [0–1]. De-normalization is the inverse function of normalization. Re-normalization is applied when a new value arrives which is outside of the existing interval. It is necessary to re-normalize all the values previously normalized with the old interval.

10.6.3 Evolution of Knowledge

The algorithm of generalization creates new hypotheses, rules, and concepts that become new words in vocabularies. The same algorithm is applied to its own results. Now the previously generalized experiences are generalized again, and the hypotheses and rules of ``lower resolution" are obtained. This evolution of acquired knowledge is illustrated in Figure 10-7.

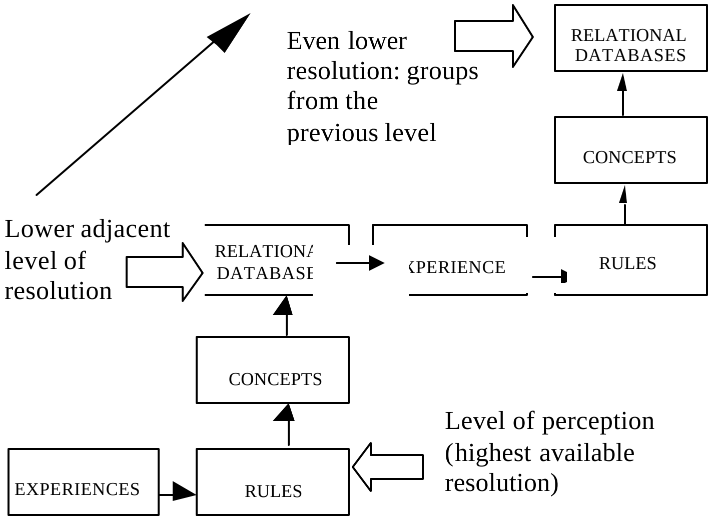


Figure 10-7 Evolution of Acquired Knowledge

This diagram shows that the joint system of knowledge representation and behavior generation converges to a multiresolutional one. This convergence is completed, only after many new concepts are obtained and the consecutive generalization processes run a sufficient number of times. Then, formation of levels can be obtained by clustering of the multiplicity of outcomes of these consecutive generalizations.

10.6.4 Focusing Attention

This procedure is applied to the complete database of experiences. It outputs the set of admissible experiences. This is a procedure of focusing attention. At this stage, we narrow down the bulk of available data using the most general similarity feature, their ``goodness" or ``badness."

There are different options in selecting experiences suitable for generalization:

1. Select good/bad experiences which are close to each other in their value of goodness:

   a) experiences with the value of goodness higher/lower than a certain threshold.

   b) experiences which are a part of a particular top/bottom fraction of best/worst experiences (percentage threshold)

   c) experiences which form a group of n best/worst of them (quantity threshold).

2. Select good/bad experiences that are close to the current situation:

   a) experiences which are closer than a certain threshold to the current situation

   b) experiences which are the part of some top/bottom fraction of closer experiences

   c) experiences which form a group of n closest ones.

3. Intermediate strategies are possible. For example, if option 1 is applied and doesn't find any rules with recommendations about the current situation, then option 2 can be applied. Option 2 can be used with different thresholds. If a large value of a threshold of closeness is chosen and no rules are found, then, for the current situation smaller threshold should be made. If no rule has been found after a few iterations with option 2, then, the learning algorithm should collect more experiences about the current situation.

When the database of experiences is large, the percentage threshold may choose too many experiences. A reasonable approach seems to be to select the ``n" best experiences. In this case, we avoid setting a minimum value of goodness while retaining the ability to set the maximum amount of computational burden for the learning algorithm.

At the stage of combinatorial enhancement, focusing attention is applied by using some selected experiences to create enhanced situations and other to create enhanced actions. In this Section only addition and subtraction were mentioned for constructing enhanced situations and actions. Other operations can be used and they will generate larger sets of rules.

10.4.5 Formation of Similarity Clusters

In this subsection, we describe a process of grouping previously chosen good/bad experiences into clusters of similarity. The procedure takes as an input the sets of experiences and enhanced experiences created in the previous steps and finds inner clusters among them. As shown in Figure 10-8, it inputs a set of groups of selected experiences and outputs clusters.

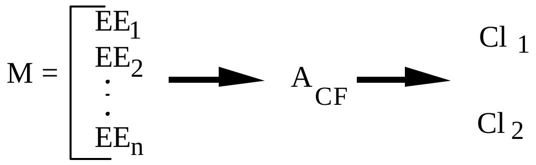


Figure 10-8 The Class Forming Algorithm

Clustering allows for reduction of computational complexity because of the following reasons:

• The system cannot store all particular experiences; creation of group allows for compression of data.

• Many of experiences are ambivalent; the system needs only those experiences that can imply a set of actions or restrictions

The following requirements should be satisfied for the set of experiences used by clustering:

1. There are no repeated experiences in any of the output clusters

2. Every experience at the inputs should be included to one of the output clusters, i.e. the cluster forming algorithm does not eliminate any experience

A. No new experience are generated by the clustering algorithm

The comprehensive arsenal of theoretical tools for generating clusters can be found in [11]. We have explored two simple approaches for formation of classes: the ``jump" approach and the closest neighbor approach. More sophisticated approaches are known [12, 13]. We would expect that the proper algorithm of clustering should be chosen taken in account all multiplicity of factors presented in the description of the environment within which the learning system is functioning.

The ``jump" approach is similar to coordinate-by-coordinate approach introduced in [14]. Instead of ``sparseness," we would prefer to use the term ``density of good (or bad) experiences." Separating clusters by using the jump threshold leads to a large percentage of meaningless recommendations if clustering is performed on all experiences. However, in the initial set of experiences, only the good (or bad) ones are represented. So, the reason that we find jumps is because of inner cause-effect correlations. Only certain actions should be applied from the whole repertoire of possible actions. These actions can only be applied in certain clusters of situations to give acceptable (or unacceptable) values of goodness.

This will cause the algorithm to find rules of action only for unequivocally advantageous (or totally unacceptable) outcomes that should generate positive and/or negative rules. If no applicable rules can be found in a certain situation, then a new goal must be declared. This assigns a situation for which a good rule was found as a subgoal. Under this subgoal, a rule will be found or again a new subgoal will be declared.

The jump threshold is a measure of density of good experiences in the domain of a particular variable, because it separates the classes by means of finding domains of space that separate places where the good examples are situated densely. Most of the clusters separated by a jump become parts of meaningful rule hypotheses.

Another approach to clustering is merging based on a minimum distance or ``the closest neighbor'' method. Its strategy is to merge the most similar pairs of experiences at each step. The criterion used to select the closest experiences is similar to the idea applied in the stepwise optimal hierarchical methods. This makes the smallest possible stepwise increase in the sum of squared errors which is also a very common algorithm for finding clusters.

10.6.6 Searching for Valid Hypotheses among Clusters.

The reason behind the creation of clusters is our belief that each of these clusters is a candidate to become a rule of a different behavior. Also, we assume that there exists a class of enhanced representation actions A_{Enh} which has already been applied to a class of enhanced representation situations S_{Enh} and which produces values of goodness in the interval from J_{min} to J_{max}. If we have a situation belonging to a class S_{Enh}, then an action can be determined within class A_{Enh} which will provide goodness J_{max}. Then, each class of enhanced experiences becomes a hypothesis.

The hypotheses are then stored in the database of hypotheses as a tree where each hypothesis is related to its ``parent'' by the goal. If no hypothesis was found for a certain situation, then the situation in the hypothesis with closest situation to ours becomes a subgoal. This is one more source for the emerging hierarchies of acquired knowledge.

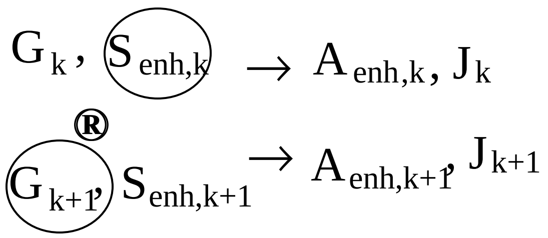

G k , S enh,k → A enh,k , J k
®
G k+1 , S enh,k+1 → A enh,k+1 , J k+1

Figure 10-9 Parent and child hypothesis

Figure 10-9 shows a parent hypothesis at the top, and a child hypothesis takes its goal from the parent's situation at the bottom. Since the child hypothesis has a new goal (which is the parent's situation) it has a new measure of goodness.

It is not possible to find rules for all situations if only good experiences are selected to the classification procedure. Other parts of the space will use as a goal the situations with existing ``direct'' rule of action toward the original goal. Two cases of hypotheses formation are to be considered: based upon situations (case A), and based upon actions (case B).

Case A.   Suppose that two or more hypotheses in the database of hypotheses have the following characteristics:

1. They have the same goal.
2. They have the same enhanced action set.
3. They have different enhanced situation sets.

Then the following options exist:

1. Their situations do not intersect. In this case, there are the following options:

   a) to leave them as separate hypotheses

   b) to create a joint hypothesis $\{S_{Enh1}, S_{Enh2}\} \rightarrow \{A_{Enh}\}$

2. They intersect, which again gives two options:

   a) to create a new joint hypothesis with situation $\{S_{Enh1}, S_{Enh2}\}$

   b) to create a new joint hypothesis with situation $\{S_{Enh.new}\}$.

Case B.   Suppose that we have two or more hypotheses in the database of hypotheses that have the following characteristics:

1. They have the same goal.

2. They have the same enhanced situation.

3. They have the different enhanced action.

In this case it is necessary to merge their actions which produces two cases:

a) if their actions do not intersect then they will be left as separate hypotheses.

b) if they do intersect, a new joint hypothesis with action $\{A_{Enh.new}\}$ should be created.

10.6.7 Learning as a Part of Behavior Generation

The second process characteristic for the learning automaton is Behavior Generation (BG): synthesizing the preferable behavior (see Chapter 7 and references [4,16-20]). This process is understood as a sequence of top-down planning activities that converge with receiving a set of control commands. The purpose of *learning* is to enable the subsequent process of *planning* as a part of BG-process. In this sub-section we talk about the process of deliberative planning based upon exploration of “imaginary alternatives” of motion. Conventional automata are only capable of reactive decisions; they are not capable of ``look-ahead'' decision making processes that are typical for deliberative planning. Unlike learning, which develops bottom-up and works via generalization (with subsequent coarsening), the process of planning develops top-down and works via instantiation (with subsequent refinement.) The process of planning is determined as choosing the desirable behavior by anticipating admissible alternatives among possible behaviors and selecting the best of them by comparing tentative trajectories in the state space (finding the planned trajectory, or PT).

Previously, we came to a conclusion that all experiences acquired and hypotheses generated contain knowledge of some reactive rules. For example, ``if it is necessary to get to S_2 from S_1, apply A_{12}.''. By following this rule, LPA reacts by evoking the action A_{12} in response to the need of getting into S_2 from S_1 at the i-th level of resolution. As this rule is applied top-down, it turns out that the space between $S_{1,i-1}$ and $S_{2,i-1}$ contains more nodes $S_{j,i-1}$, where j=1,2,3... are nodes of (i-1)-th level of resolution. The process of selecting any string from them requires deliberation. Thus, multi-step search procedures were recommended in [14].

PT is a trajectory which satisfies the specifications (the latter are determined by the Goal). Trajectory is a string of adjacent admissible elementary subdomains (or tessellata, tiles of the discretized space) denoted as $\{W_{ij}\}$ where i is the level of resolution and j is the number of the tessellatum in the string $\{W_{ij}\}$. If necessary, PT can be also represented as a sequence of the subdomain indices $T_{1\{j(m)\}}$

where m- is a number of elementary subdomains in a string {m=1,2,...,z}. In this Section, we address only a well-posed problem (wpp) of planning. Any wpp of planning should: a) start with assigning the initial point (SP_{i-1}) and the final point (FP_{i-1}) of the trajectory which should be determined at the higher resolution (i-1) within the tessellatum of the resolution under consideration (i), and b) the feasible trajectory should be determined first at the lower level of resolution (i).

PT is called a feasible trajectory if it has an initial point $SP_{i-1} \in \Omega_{i,1}$, a final point FPi-1 $\in \Omega_{i,n}$ and all tiles of the string are contained within the envelope formed by the feasible trajectory at the lower level of resolution (see Chapter 8). Thus, a feasible trajectory for the level i is always represented as a string of tiles for the i+1 level of resolution and is a subspace of the i level of resolution. We determine the subspace in which the wpp of planning should be resolved, and the optimum trajectory should be found. Indeed, the feasible trajectory determined at the i+1 level of resolution becomes an ``envelope," a bound domain of space at the i level of resolution.

The recursive definition of the feasible trajectory does not lead to any infinite incomputable computational procedure. As we extend our search for the feasible trajectory to the consecutively lower and lower levels of resolution, we eventually arrive at a level where both the initial point SPi and the final point FPi belong to the same tessellatum of the lower level of resolution: FP_{i-1}, SP_i, $FP_i \in \Omega_{i+1}$ which is the initial space for a feasible trajectory for the i level.

The sequence of feasible trajectories can be considered a nested system of spaces for multiresolutional search. To increase the reliability of searching for optimal trajectory, we increase the envelope of search by surrounding the feasible trajectory by one or more adjacent tiles along its boundary that will determine a particular ``width" of the envelope. This results in a system of enhanced volumes $V_1 \supset V_2 \supset ... \supset V_k \supset ... \supset V_m$ which we will use later for searching. The optimum trajectory at the i level of resolution is a minimum-cost path in the graph which is built upon all center-points of the tessellata at the i level of resolution. These tessellata are constructed within the envelope formed around the feasible trajectory found for the i+1 level of resolution.

The definitions for feasible and optimum trajectories imply the off-line method that has been introduced to find the best trajectory of motion to be followed by the control system. Search in the state space (S^3-search, see Chapter 8 and references [8, 14-20]) is done by synthesizing the feasible trajectories for the i level of resolution and then building the alternatives of possible motion trajectories for the i-1 level within the envelope cost space.

Some particular volume of the state space is designated for a subsequent search for a solution, Operation of contraction puts constraints on this volume and should be properly justified. We need to reduce the probability (the risk) that contraction eliminates some or all of the opportunities to find the optimum path trajectory. The following heuristic strategy of contraction is chosen. After the search at the lowest resolution level is performed, the optimum trajectory is encompassed by an envelope. It is a convex hull which has a width w determined by the context of the problem. Then, the random points generation at the next level of resolution is performed only within this envelope of search. This strategy is demonstrated to be acceptable in many practical cases. However, the problem of consistency of representation under the contraction heuristic has to be addressed in the future.

LPA merges the deliberative planning with learning. Both are procedurally kindred and match from the software engineering point of view. They produce and use the same multiresolutional system of representation. Both use the same Ω-state space in which the start and final points SP and FP are given. The path from SP to FP is to be found with the final accuracy ρ. Operator of Learning L (Ω,ρ) constructs the system of representation via creation of generalized maps $M(\Omega,\rho)$ at each level of resolution and obtained by generalization of the higher resolution map (ρ corresponds to the level of resolution of this map determined by the density of the search-graph).

Even if the learning automaton was not given any concept of ``self," this concept will be developed automatically by the algorithm of learning. At the lower resolution level, the deliberation of strings is expected, combined from cells of the higher level of resolution. Concatenation of these strings is possible only via concept of the ``current state." It is possible to demonstrate that this concept will evolve into the concept of ``self." Since the complexity of computations is drastically reduced when both the surrounding objects and the LPA are represented in the same map in some global system of coordinates. Thus, learning automaton will be able to put itself in a map.

10.6.8 Evolution of Multiresolutional Learning Automata

Figure 10-10 shows a detailed, enhanced version of the elementary loop of functioning (see Chapter 2 and references [2,3]). This is a symbolic representation of the processing during various learning activities typical for a system equipped by modules unsupervised learning (UL) and behavior generation (BG). This remains the same in all goal-oriented cases of automata equipped with the joint ULBG modules. Perception allows for recording the set of recent experiences in a symbolic form. By grouping the experiences, the classes of similarity are discovered. This induces the hypotheses explaining the similarities, or instigates new experiences belonging to the same class of similarity. Within the semiotic paradigm, the loop of ULBG can be called ``a loop of semiosis" [2].

The system is presumed to function under an externally assigned Goal. The initial set of experiences (which might be obtained by random actions) is generalized into hypotheses. The hypotheses enter the subsystem of Behavior Generation as a substitute for the rules, The decision for an action is made; the action is performed; changes in the world occur; the transducers (sensors) transform them into a form that can be used by Perception. The long and complicated process of moving from signs to meaning starts again. Now, the enhanced set of experiences brings about another hypotheses that can confirm or refute the tested ones. This is when the symbol grounding happens.

After multiple tests, the hypotheses can cross the threshold of "trustworthiness," and a new rule is created. A rule (or a set of rules) within a context is considered to be "a theory." At each development step, the unit under consideration undergoes a comparison with other kindred units confined in corresponding databases (of Experiences, of Rules, and of Theories.) Then, the symbols tentatively assigned to some "unities," "entities," or "concepts" enter their place within the database of concepts (which is a relational network of symbols.)

Rules (or the hypotheses that will become rules) are formed when experiences cluster together unified by their similarity. For the prior state S_1 the applied action A_{12} leads to the emergence of the new state S_2; the value of reward J_{12} is the result. After gathering a sufficient number of the experiences and generalizing them properly the rules of the following form can be constructed: *IF the value J is desired upon achievement of a the goal-state SG from the present state S1, THEN the action A1G should be applied*.

An interesting and unique feature of generating rules can be described as follows. Each component of a rule is a generalized component of experience. This means that to obtain a component of a rule, several similar components of experiences should be grouped together into a class, a cluster. This requires applying a set of procedures using the triplet of grouping, focusing attention, and combinatorial search. The label attached to this cluster signifies the process and the result of generalization. The premises behind the process of generalization could be different. But, the result will be always the same: creation of the new object for the lower level of resolution.

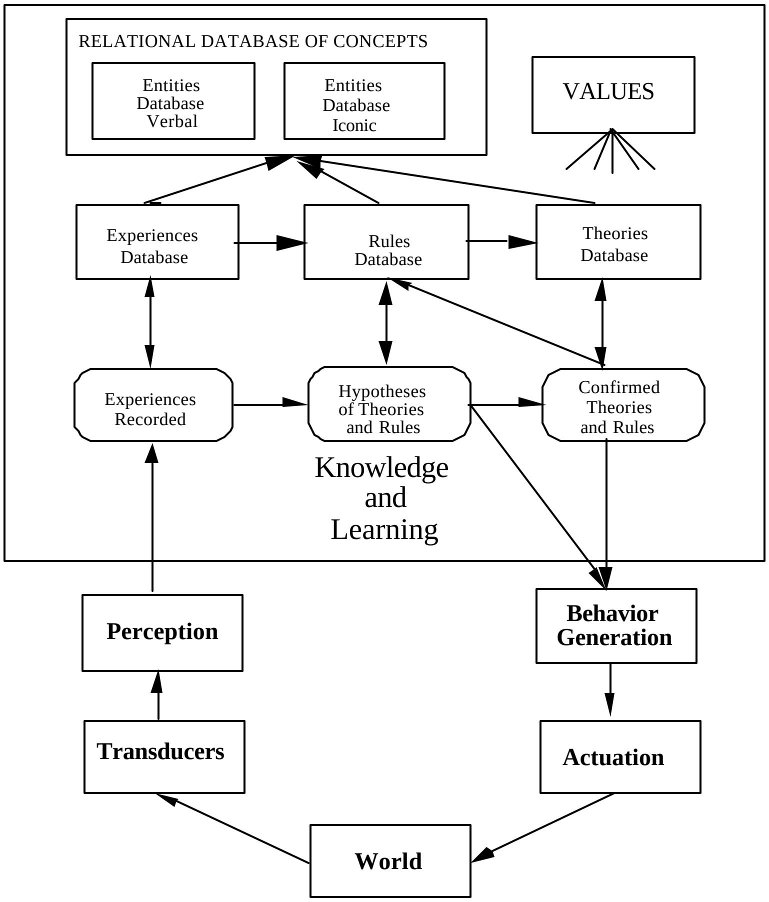


Figure 10-10 Functioning of Learning Automaton: ELF with Learning

For example, let G_i symbolizes the phenomenon of generalization upon i similar experiences (i=1, 2, ..., n), then

(9) [The zone of states ``S" with $J_1<J<J_2$] → $G_i\{S_i, J_1<J_i<J_2\}$,

(10) [Action ``A" to be applied to achieve a desired zone] →Gi{Si, Ai,i+1}.

Only the desired state is not subject to generalization. It is always individual, pertaining to a concrete system and problem.

10.6.9 Further Research in LPA

Theoretical analysis presented in this Section has been confirmed by the experimental results [1,7,8]. ``Baby-Robot" was able to learn how to reach the arbitrary situated goal only after implementing the algorithm of generalization with combinatorial enhancement (see Section 10-7)

Before this algorithm was implemented, Baby-Robot was able to learn how to reach the particularly situated goal. If the location of the goal was changed, the successful learning process for the previous goal could not help to find a new one. Generalization n with combinatorial enhancement has enabled the robot to make the discovery, and initiate the process of hierarchical learning. Other positive results are recorded in [19,20].

1. An algorithm of multiresolutional unsupervised learning with inductive generalization and a search for hidden implications has been introduced and tested. This algorithm is applied recursively to its own results at the output. Therefore, it builds up a multiresolutional structure of results. This type of unsupervised learning (UL) is demonstrated to be a mechanism of development of an evolving multiresolutional system of representation. It enables the system of behavior generation (BG) also to evolve. Together, (ULBG) they provide for an evolution of the automata equipped with such systems. The automaton equipped by ULBG becomes a multiresolutional automaton, and its levels of resolution can change as the evolution of knowledge and behavior proceeds. This evolution can be illustrated by Figure 10-11.

2. From Figure 10-11, one can see the behavior evolves. This producing different plans and motion trajectories as shown as a horizontally developing tree. At the same time, its knowledge evolves as shown in the vertical hierarchical structures. This process seems to be even more important to analyze of the evolution of living creatures. We believe that the automaton equipped by ULBG allows for analysis of the processes of evolution of these systems (automata with ULBG) as species. It is possible to equip the automaton by the system of reproduction. It would be possible to analyze how the process of knowledge evolution is affected by different mechanisms of reproduction.

3. This line of research takes advantage of the uniqueness of automata with ULBG among other known systems of automata with learning. The mechanism of unsupervised learning allows for the ultimate freedom in the way the learning process organizes the acquired knowledge. It is possible to anticipate that as the knowledge base evolves, the knowledge becomes utterly diversified. Rules concerning the external world will emerge, and the rules concerning processes of inner knowledge organization and procedures of processing will follow. Simple mental experiments can confirm that the system which starts with perceiving the world as a set of values {vij}, i.e. having an ``ego-focused'' representation will learn how to develop maps in externally-fixed coordinates that will allow for putting on the map the system itself (which might be interpreted as emergence of ``consciousness'' in some

applications).

4. The automaton with ULBG can be used to analyze all stages of learning including the ``early learning'' stage. Certainly, some initial knowledge (``bootstrap knowledge'') is presumed. This bootstrap organization of knowledge can strongly affect the subsequent processes of knowledge evolution. On the other hand, the learning system is presumed to be free of a building up of all subsequent knowledge organization. How it will organize the knowledge acquired and why— this is the research issue for the are a of automata with ULBG.

5. The processes of knowledge acquisition are affected by the knowledge stored. They start creating some bias in the subsequent knowledge acquisition since the results of automaton functioning will be induced by the knowledge previously stored. So, if the results of functioning were ``good'' or led to a ``better'' behavior, the system might assume that its goodness is due to the knowledge used. It might happen that the experiences the system acquires are limited by its predisposition. This is another research topic of interest in the ULBG area.

6. Since the system has been developed to demonstrate some particular behavior, the evaluation of this behavior should be the ultimate measure for the process and results of knowledge organization as well as the processes and results of the ways knowledge is acquired from the external world and knowledge used for behavior generation. This determines an interesting interconnection between further developing the hierarchy of functions for evaluation of goodness which precipitate at different levels of resolution. The interconnection between learning, behavior, and developing of the ``system of values'' in the automata with ULBG seems to be an important research issue.

7. In concert, all these three processes: acquisition, organization and use affect the overall system functioning. Therefore, other systems of learning can modify and alter the system of ULBG.At the present time the following mechanisms of knowledge acquisition are known from the literature (other than UL):

a). Learning by transfer. In this case all knowledge which subsequently is required for behavior generation is transferred from another source where it was stored and organized in advance based upon existing design decisions and experiences of functioning. This method of knowledge acquisition presumes that the structure of the system of interest and its functioning in required circumstances are previously known, and knowledge is organized so that this structure be properly supported.

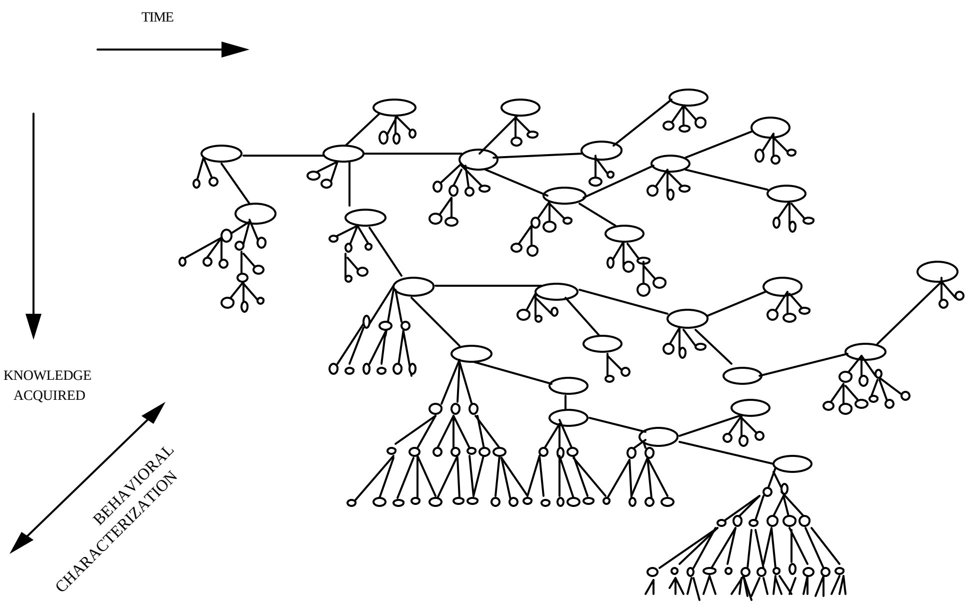


Figure 10-11 Evolution of Knowledge and Behavior of the Automata with ULBG

b). Learning by examples. In this case, we presume a ``Teacher'' which has substantial knowledge about my cases of possible functioning, stores knowledge of previous experiences and spells out a set of possible scenarios in which the functioning of the system is expected. Undoubtedly, a set of tests can be developed in which a behavior of system is entertained and after each case of behavior the system receives the teacher's evaluation whether it was good, and how good it was. These tests teaches exercises for which the solution is known. Interaction of ULBG system with other systems of learning should be a separate research issue.

The system is to be taught the responses to the test by demonstrating a particular behavior. At the end of the test, it is informed by the teacher whether this behavior was right or wrong. In more complex schemes, it can be informed of how good it was and why. Unlike in the mechanism 1, the mechanism 2 does not interfere with how the learning system organizes the required knowledge. However, the teacher's interference into the process of knowledge acquisition can be deep enough. The mechanism 2 does not say anything about the way knowledge should be organized within the learning system.

8. Learning and Behavior Generation produce structural and behavioral hierarchies. It is possible to state that using ULBG reduces computational complexity by increasing the structural complexity.

References to the Section 10.6

1. Meystel A. Baby-robot: On the analysis of cognitive controllers for robotics, Proc. IEEE Int'l Conf. on Man and Cybernetics, Tuscon, AZ, Nov. 11-15, 1985, pp. 327-222

2. Meystel A., Intelligent Systems: A Semiotic Perspective, International Journal of Intelligent Control and Systems, Vol.1, No.1, 1996, pp. 31-57

3. Albus J., Meystel A., A Reference Model Architecture for Design and Implementation of Semiotic Control in Large and Complex Systems, Proc. of the 1995 IEEE ISIC Workshop, Architectures for Semiotic Modeling and Analysis in Large Complex Systems, Monterey, CA 1995, pp. 33-45

4. Meystel A., Nested Hierarchical Control, in Eds. P. Antsaklis, K. Passino, An Introduction to Intelligent and Autonomous Control, Kluwer Academic, Boston, MA 1992

5. Rajaraman K.,Sastry P.S., Finite Time Analysis of the Pursuit Algorithm for Learning Automata, IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, Vol.26, No.4, August 1996, pp. 590-598.

6. Albus J., Outline for a Theory of Intelligence, IEEE Transactions on Systems, Man and Cybernetics, Vol. 21, No. 3, May/June 1991, pp. 473-509

7. Albus J., Lacaze A., Meystel A., Autonomous Learning Via Nested Clustering, Proc. of the 34th IEEE Conference on Decision and Control, New Orleans LA, 1995, pp. 3034-3039

8. Albus J., Lacaze A., Meystel A., Theory and Experimental Analysis of Cognitive Processes in Early Learning, Proc. of the 1995 IEEE Int’l Conference on SMC, vol. 4, Vancouver, BC, Canada, 1995, pp. 4404-4409

9. Fogel D.B., Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, IEEE Press, New York, 1995

10. Meystel A.,Thomas M. Computer aided conceptual design in robotics, Proc. IEEE Int'l Conf. in Robotics,

Atlanta, GA, March 13-151984, pp. 220-229

11. Mirkin B., Mathematical Classification and Clustering, Kluwer Academic, Dordrecht, 1996

12. Mirkin B., The Method of Principal Clusters, Automation and Remote Control, 1988, pp. 1379-1388

13. Mirkin B., Method of Fuzzy Additive Types for Analysis of Multidimensional Data, Automation and Remote Control, 1990, pp. 683-821

14. Michalski R.S, A Theory and Methodology of Inductive Learning, Artificial Intelligence, Vol. 20, No.2, Elsevier, Amsterdam

15. Goldfarb L.,Abela J., Bhavsar V.C., Kamat V.H. Can a Vector Space Based Learning Model Discover Inductive Class Generalization in a Symbol Environment, Pattern Recognition Letters, V. 16, 1995, pp.719-726

16. Meystel A., Planning in a hierarchical nested controller for autonomous robots, Proc. IEEE 25th Conf. on Decision and Control, Athens, Greece, 1986

17. Meystel A., Planning in a Hierarchical Nested Control System, in Eds. W. Wolfe, N. Marquina, Mobile Robots, Proc. of SPIE, Vol. 727, Cambridge, MA 1986, p.p.42-76

18. Saridis G., An Integrated Theory of Intelligent Machines by Expressing the Control Performance as Entropy, Control: Theory and Advanced Technology, v. 1, No. 2, p.p. 125-138, MITA-Press, Tokyo, Japan, 1985

19. Meystel A., Uzzaman S., Landa G., Wahi S., Navathe R., Cleveland B., State Space Search For An Optimal Trajectory Proc. of the IEEE Symposium on Intelligent Control, Vol. II, Philadelphia, PA, 1990

20 Grevera G., Meystel A., Searching for a path through Pasadena, Proc. of the IEEE Symposium on Intelligent Control, Arlington, VA, 1989

20. Quinlan, J. R., “Combining Instance-based and Model-based Learning”, In Proc. Of the 10th International Conference on Machine Learning, Amherst, MA, 1993, pp. 236-243

10.7 BABY-ROBOT: Analysis of Early Cognitive Development

The Section is dedicated to the processes of Early Cognitive Development (ECD) in Intelligent Systems. The similarities and differences are stated, existing at the stage of ECD of humans and robots. It is shown that there are two types of learning: quantitative (QL) and conceptual (CL). The role of QL and CL in ECD is discussed. The concept of Baby-Robot is devised as a tool for investigating ECD processes. Baby-Robot is expected to be instrumental for the analysis of mechanisms of autonomous knowledge bases generation. The structure of Baby-Robot is a seed structure for building up the self-organizing cognitive processes in Intelligent Systems.

10.7.1 ECD: Its Significance for Learning in Intelligent Systems

The problem of Early Cognitive Development (ECD) appeared due to the new areas in Intelligent Systems (IS) including robotics and automation, systems for intelligent control, autonomous mobile systems, unmanned industrial devices, and even unmanned systems of integrated manufacturing. All these areas of application of IS are characterized by the following distinctive properties:

a) IS is given a “goal” which must be achieved in the process of the IS functioning. This goal is given not only as a description of the required changes in the world but also as the cost-functional to be maximized (or minimized) in the process of operation, and as a set of constraints to be satisfied during the operation. The “operation” is understood as a combination of actions that are to be undertaken in order to achieve the goal. In this Section we will consider only one class of IS: the one that achieves the goal by developing a specific motion trajectory.

b). The notion of achieving the goal “successfully” is specified in the IS vocabulary however the reality of the successful achievement can be beyond the IS “understanding” at the moment of task formulation. Realistically, IS operate

specification. Thus the operation is performed under condition that the initial information is incomplete, the model cannot be captured in all detail, the sensor information is not sufficient, contains errors, and the means of actuation are limited.

The distinctive properties mentioned above are typical for most of situations in which the application of unmanned autonomous intelligent robots is expected and desired. Clearly, the existing concepts of control based upon hierarchical systems of intelligence [1,2] and using knowledge-based controllers to generate execution commands [3] are supposed to be utilized for intelligent robots in general, an for Intelligent Mobile Autonomous System (IMAS) in particular. However, the system of knowledge required for actual operation of these devices seems to be enormous.

Even in a case of limited operation (e.g. computer simulation of a 2D world with a multiplicity of spatial relationships in a system "IMAS-Set of Obstacles") the authors came to the conclusion that it is impossible to imagine and put in the knowledge-base all situations that can generate meaningful rules [3]. IMAS must be able to recognize that some of the unexpected spatial situations. Interestingly enough, in order to recognize the "trap" situation and a corresponding rule of action for a particular motion trajectory, IMAS should remember the trajectory of its motion during at least some past limited time period. Thus, learning from experience is a vital part of dealing with the reality even for a simulated IS.

Therefore, the recognition of familiar situations stored in the memory is not sufficient. Indeed, not in all of the "trap" or "difficult" situations IS start moving "in circles" and repeat themselves: this is just the easiest case. It would be desirable to have an opportunity to ***learn from an experience with no prearranged patterns given***. The number of situations which the IS will encounter is expected to be immense, all their patterns cannot be given to the IS "at birth". In this Section we will explore a different approach: IS must be capable of learning the situations of different kind, and the solution of the problem should be found depending on the kind of the present situation given a particular goal. We will consider only one class of robots: which are at the initial stage of their existence and do not have neither a teacher, nor any prior experience.

This Section addresses the problem of designing the structure of IS with this particular capability: early learning. Resemblances and differences between the ECD processes in both humans and IS are discussed. ECD structures are surveyed, the difference between quantitative and conceptual types of learning (QL and CL) is outlined. The structure of Baby-Robot is described.

10.7.2 On the resemblances and differences between ECD in human and ISs.

Resemblances. A limited repertoire of unlearned behaviors can be given (and is given) "at birth" in both cases. These behaviors can be considered as a limited list of actions-primitives ("suck", "grasp", "blink", "accelerate", "turn", etc.). Piaget labels the primitive behavior by term "schema". Schema is understood as an elementary unit of the cognitive structure that in many cases is understood as an activity A with its structural connotation.

Each of these clauses in understood as a concept of behavior (C_B). Another resemblance is related to the so called "sensorimotor" activities which are based upon direct chain between sensation and physical movement which presumably does not include complex forms of cognitive activities such as "imagination" and "problem solving". Thus, sensorimotor schemata can be written in a form of clauses. Structural connotation of clauses is not explicitly recognized by an actor, and does not participate in the control of the sensorimotor activities. This representation does not contradict to the existing views of the scholars in the area of cognitive development [4-6]. As one can see, ECD can be described adequacy enough in the framework of the automata theory. This corresponds to the closed loop theories of motor learning [5]. Thus a set of "senses" is also presumed as well as the set of above mentioned primitive actions. Based upon these two sets, a set of rules is formed where each of the rules is represented by the clause.

Psychologists are used to talk about logical weakness of the child's reasoning which is said to be syncretic and transductive. Syncretic reasoning reflects processes of classification when the criterion of classification is changing in time. Transductive reasoning (from particular to particular) is a supplement to more common types of reasoning: deductive (from general to particular) and abductive (from particular to general). At the initial stages of ECD, a human child is characterized by egocentric mentality, perception is the dominating way of thinking. An IS at the ECD stage (Baby-Robot) is expected to define its state in relation to surrounding objects and vice versa. As the system evolves, Baby-Robot develops the need in a capability to represent the world in some global coordinates. This ability evolves as it is being developed in human child. The capability to choose the most appropriate reference frame should be given at birth as one of the important "seeds" of the future development. It seems, that for the functioning GFACS, this capability is a byproduct.

It is expected that for the Baby-Robot, most of the situation (S) and goal (G) description should allow for a translation into set of Rules [R] for which the following holds: $\{[R]\wedge S\wedge G\}\rightarrow A$. These implications are to be developed during the "teaching tests" which are analogous to the sensori-motor play in a human child. It is possible also to expect that for a number of problems, the analog for so called "imaginative plays" can be also found.

The distinction between exafference and reafference in the Baby-Robot is the same as in a human child[51]. Since it was shown that learning based upon individual's own movements is important for the development of accurate visually guided behavior [8], the strategy of programmed procedures of reafference are especially important for the Baby-Robot.

[51] Motion control of animals is considered a feedback loop. It works as follows. The *afferent* nerves carry signals toward the central nervous system while *efferent* ones carry signals from the central nervous system to the motor areas. Afferences are divided into receptor excitations of two types: *reafference* caused by internal changes in the musculature and *exafference* produced passively by external stimulation.

Differences. One of the major differences was mentioned above: a human child is egocentric initially. A baby-robot is not necessarily egocentric, many different frames of references are easily given to it at birth. Many of the general postulates of ECD that are discoveries for a human child, could be given to a baby-robot in a "burnt-in" form (such like rules of conservation, reversibility, identity, and ability to create combinations). Hence unlike a human child, the baby robot might not have a period of "proportional thought", and the ability to perform concrete and even formal operations is suppose to be given to the baby-robot at birth. The same is related to a number of reflexes and responses that might be conditioned at birth if the manufacturer chooses so. The same is related to the infant states including alert situations, and focused activities.

Most of the differences are related to the processes of development linked to communication. Since we limit out analysis by consideration of a single IS, the social play is not expected to be reproduced. (The problem of robotic team is not discussed here.) However, the most important difference is linked with the drive to activity. IS is typically built for use in a definite spectrum of goal-oriented activities. In other words, the hierarchy of tasks can be always formulated and the world description can be always structured with respect to the hierarchy of tasks, and the spectrum of available means of performing these tasks. Certainly, we can consider human behavior within a task-focused paradigm ("task-driven behavior".) However, it is impossible to do so when ECD is analyzed. Indeed, baby-robot is given its task vocabulary at birth whereas a human child cannot condition his or her behavior by a clearly cognized task in most of the ECD situations.

One of the major general differences which at the present time seem to be out of reach for the scientific effort, if the innate ability of humans to organize input information according with the context and meaning. The rules of "gestalt" might be programmed but this cannot ensure their proper application within the structures of learning, and problem solving unless we know the mechanism of their development (possibly in the form of GFACS-triplet).

10.7.3. Quantitative Learning Domain.

Frequently, the "knowledge" of IS as understood as something that is to be given at the beginning [9]. Nevertheless, the need in learning as a mechanism for on-line knowledge acquisition was recognized, and in the 60's the active efforts were undertaken. One of the first accounts of a learning systems is given in [10] where the results are given of the earlier works in the area of learning. The concept of learning network quickly became dominating one [11]. A simple learning system is described in terms of the four component-networks: sensors that provide data for learning networks (LN). LN responds to the input (stimulus) by developing the output (response) which generates the commands for actuators. Their action is evaluated by the goal network that generates either reward or punishment. In [12] the same diagram is disclosed in more detail. The elements of the theory of learning systems are attempted in [13].

Adaptive neural network concepts such as “perception” [14,15] or “adaline” [16] perform supervised learning pattern classification. They form their discrimination rules, if the “teacher” can supply each component adaptive element with its individual desired response. It was not immediately recognized that the learning process within the adaptive network, associative network, percept on, etc.) does not create any new concepts, just a quantitative adjustment takes place. We will name this process “quantitative learning” (QL).

On the other hand, after process of learning is completed, each elementary particular network can be considered as a particular “concept”. Using the neural network (NN) for concept formation is proposed in [17]. Finally, it was recognized that after the learning process in the NN converges, the result of learning must be symbolized by a new label corresponding to the new concept [18]. We will name the process of learning a “conceptual learning” (CL) if the system of concepts is obtained as a result of learning.

In the meantime the processes of QL has been developed substantially and gained many important properties and high level of sophistication. In [19] NN is treated as a kind of perception, and shows that the learning process converges for a large class of distributions. The intention to provide for a learning process without a teacher gives some positive results [20]. Using NN for control of autonomous mobile systems is described in [21, 22].

Authors of [23-27] have proposed “associative search element” (ASE) as well as “associative search network” (ASN) which also learn from their experience, however they do not require a teacher. They generate actions by a random component of the generation process introduces the variety that is necessary to serve as the basis for subsequent selection by evaluative feedback.

Contemporary models of multilayered perceptrons are representing the most advanced concepts of hierarchical world representation in visual domain [28]. These models might be a source of a concept generating structures. However, this problem is beyond the focus of this Section. A number of applications of NN is known in the area of manipulator control [29,30]. In all of the known works QL based upon” teaching tests” is described. A good example is given in [32] where after the repeated trials of a path, the compensation is achieved for effects such as friction and torque due to velocity and gravity effects. QL is applied within a number of intelligent systems. Usually, it is based upon changing values of the coefficients in the pattern classifier as a result of statistical learning under the accepted reinforcement law and accepted technique of memory adjustment [38].

It is important to underline the following fact: QL demonstrates two general concepts found in the experience of living nature. One of them is based upon using statistics to induce a tendency in change [32]. Another is an idea of choice since the discriminators based upon various decision rules are utilized in the system of QL. Assuming a probabilistic description of the uncertainties, a classical decision rule is to choose the strategy which minimizes the expected cost. It was noticed in [33] that this way did not reduce the probability of getting a bad performance in the realization, and a decision rule was proposed which consists of minimizing a combination of the expectation and variance of the cost [34].

Based upon these two ideas, the method of random search has been created, (one of the first instrumental descriptions applied to learning processes is given in [35].) In practice using the multivariate probability distribution is required for a "thoughtful" guided search [36]. Most of the genetic algorithms and systems with evolutionary programming fall under this rubric.

10.7.4 Conceptual Learning Domain.

After the process of concept formation is completed by the algorithms of classification and clustering (see Section 10.3-10.5), or after the particular NN representation for this concept can be "frozen" and transformed into a single neuron of a "higher order" the process of Conceptual Learning can be considered completed. In fact, NN are capable of learning the Boolean functions [37]. These concepts represented in the form of Boolean functions describing hierarchy of action primitives are to be combined in response to a task [39].

QL systems are also based upon a principle of inductive learning to building programs which operates in two phases [40]:

1. The Training Phase. The system decides on-line by interacting with the sensors which motions are to be executed. By performing several instances of the same task it generates multiple traces of execution.

2. The Induction Phase. The system applies transformation rules to the traces generated by the training phase, and builds a manipulator level program including symbolic variables, conditional statements, and lops.

Induction as the process of inferring the description of a class from the description of some individual of this class, was an object of attention in AI for quite a while [41, 42]. In [41] learning is understood as filling in the prearranged boxes of possible relationships among the entities of the vocabularies which are not subjected to change. However, the actual application to motion control was first presented in [40]. A typical motion statement in LM-language is "move F by T until C", where F- is an object to be moved, T-is a transformation by which F is to be moved, C-is a sensory based condition on which the motion is to be stopped even if T is not completely achieved. All of the planning rules are of the from <conditions→ <plan> and all of the execution monitor rules are of the form <conditions>→ <relation>.

The purpose of the "induction phase" is to construct an LM program from a set of execution traces which have been produced by applying the same strategies that is "by selecting the same planning rules in the same situations". The rules of induction are intended to unify in a cluster of different motions that result in the same state of different states that could be generated by the same motion.

Clearly, the state description is critical for the situation understanding (interpretation) and the plan generation. The most advanced results on the world description are given apriori in English language. The schema-based approach is presumed which is also generally accepted in most of the substantial results from [43-47].

Although, the idea of schemata-based analysis of the image (situation) has been proven to be beneficial, working with the complete image did not seem to be efficient and the idea appeared of the "focus of attention" [48]. Using the state description for the plan generation, was shown in [49]. In a multiplicity of papers the "schema-based description of robotic skills are used to generate plans [50,51].

Only recently, a number of works appeared which enable to put the CL problem in the science of intelligent systems on a level of substantially new capabilities [52,53] especially when the world representation is incomplete [54]. The following statements can be made which characterize the state-of-the-art in the area of IS motion planning and learning:

1. Training is understood a generation of multiple traces of execution.

2. The transformations of motion execution are given in advance.

3. The sensory based conditions or stopping the motion are predetermined.

4. The expert knowledge used in the process of trace generation is taken from a knowledge-base where they were placed by a user. If some knowledge does not exist the system requests for an additional rule which it does not possess.

5. All of the rules employed by the system are given in the form of implications and these implications are never questioned, they are considered as ultimate laws.

6. Strategies of selecting the same rules in the same situations are based on the assumption that the "sameness" cannot be evaluated quantitatively. In the meantime, it is not clear how to select a rule even for the completely "same" situation, if more than one rule might be recommended.

7. The logic of dealing with information ("the inference engine") is based primarily on the rules of deductive logic implanted in the mechanisms of programming languages.

The list of above mentioned properties may vary from IS to IS, and from problem to problem, however the core of it remains the same and was never challenged.

10.7.5 Baby-Robot: Simulation Tool for Analysis of ECD Processes.

In the meantime it seems to be interesting an important to consider the following problems hidden in the state-of-the-art of the IS programming decision-making, planning and learning techniques.

1. Training should be explored as "knowledge acquisition based upon intentionally organized activities" or "knowledge acquisition via redesigned experiences".

2. The list of concepts (3) not necessarily should be given in advance. One may expect that there is a minimum number of rules to be given to a IS at birth, and all other rules have to be learned (and discovered) within a definite frameworks of IS activities. In other words, the minimum list of rules-primitive is to be determined.

3. The “sensory-based conditions of stopping” (or other change of the motion mode) should not be predetermined. We are not sure that the user is always capable of giving the best solution of how to respond by selecting a particular action when the information is changing.

<i-th image j-th change>→ <A_k>

is the best rule for all of the situations the robot can face in reality.

It is expected that there should be a minimum of rules to be recommended after the change in perception is recorded.

4. Thus “the expert knowledge” to be applied for the trace generation should be generated within the IS and not suggested by a user. Furthermore, the necessity to address a user should be considered as a system failure. We expect that such failure will appear because of

a) insufficient list of rules;

b) insufficient logical capabilities;

c) insufficient mutual understanding of perception and planning subsystems;

d) insufficient list of initial rules in the “expert knowledge”,

e) natural limits for the concrete class of IS.

5. Although the form of “implication” is utilized as a major tool in the inference machines of systems for knowledge based control, it is unclear that this mechanism should bee dominating in intelligent autonomous robots. We can assume that the “implication” is not necessarily required, it can emerge within a IS in a natural way (the necessity in “implication” should otherwise be proven.)

6. “Sameness” or “degree of similarity” between two states and/or two situations cannot be assigned. The complete equivalence between two structural descriptions should be considered as an expectation rather than as a typical situation. Then the question emerges: “What is the value of structural (and quantitative) similarity which predisposes application of a definite rule?”

7. The logic of dealing with information cannot be assigned by user. The latter has created his logic (whatever it is) in different problems. A question can be asked: what is the structure of logic which should be employed by a IS? Is the predicate calculus of the first order a sufficient tool?

All of these problems stimulate the idea of Baby-Robot. It should be given capabilities of perception, conception, planning, and control actuation. However, the vocabulary of perception, conception, and motion planning and control should be acquired via operational experience. Everything should be questioned before submitting it to the Baby-Robot as “inherited information”. Indeed, the ability to generalize (to unify descriptions in a cluster by some token), should it be “inherited” or created in the process of operation? [or maybe both?] This will affect the ability to form “descriptive structures”, and to decompose them. Is the process of alternatives selection (during decision making) determined by the stored implications or it is based on ambivalent similarities? What is the relative frequency of the couple “cause-effect” which makes the couple valid for assigning it a rule status?

10.7.6. Baby-Robot: A Mental Experiment

The baby-robot behavior in a bug-world[52] can be illustrated as follows [56-59]. In Figure 10.12,a a world is shown containing three objects: a robot - R (triangle), a goal - G (star), an obstacle B (rectangle). a dotted line which connects R and G indicates the drive to achieve the goal as soon as possible.

In the beginning our baby-robot R does not know how to determine and execute ("plan and control") its own motion toward G. R decides to explore "what would happen if" and starts making tentative moves and evaluate their consequences. It initiates the first move in a random direction, and estimates the distance to G which happens to grow (position 2). This is “bad” concludes our robot, let us try something else. Another move in random direction seems to be a “good” one (position 3), another move is again “bad”. After a number of random moves baby-robot can deduce what should be done and assigns “proper direction” to the system of actuation. It is our goal to find what is the "bootstrap-knowledge" that is required to deduce the right direction from a multiplicity of random moves. It would be even more desirable to determine "the general rule" of finding direction in any situation, not only in the present state. This might require more bootstrap-knowledge". This is exactly what we want to explore.

After the process of learning has generated the concept of “direction” and the TME of motion toward the goal based upon the concept of "motion direction” starts from the position 5 toward the goal. When the arc of the “range of vision” O_1O_2 intersect the growing image of the obstacle is recorded in memory until the impact in the position 6. “Impact” can be manifested by the appearance of the second modality of perception (from the tactile or force sensors), and this modality can play a role of “pain” which can be The growing image of obstacle receives a label “bad”.

On the other hand, it is not necessary to “multiply entities without necessity” (Okham's *parsimony* principle). After impact with the obstacle the robot merely cannot continue its motion toward the goal, and other possibilities should be explored as it was in the initial position). However, now it has more knowledge and skills. It has already a concept of direction, and it knows that the obstacle is “bad”. The difference between positions 7 and 8 (Figure 1b) is not understood yet. After the trial movement toward 7 and 8, it may learn that the end 7 of the “bad” become closer while the “end” 8 remains at the same distance. The robot should link it with the limit of range of vision, thus acquiring the first evidence of this phenomena (before the statistical conclusion of the viability of some rule will be obtained). When the position 7 is achieved the completion of the “task” will be easily performed.

Some problem to be overcome, is hidden in the fact that although the “subgoal” 7 supposedly reduces the “badness” of the situation of facing the obstacle, moving toward the position 7 increases the “badness” of being remote from G. This contradiction can be better illustrated by Figure 1.c. where

[52] The term "bug-world" carries the message that for the creature under consideration the World is 2-dimensional and can be fully represented by a geometrical sketch on a sheet of paper.

moving toward 8 seem to be more preferable than moving toward 7. Indeed, although the end 7 become closer, but the "badness" of distance to the goal is growing. At the same time, although the border 8 does not become closer the "badness" of distance to the goal is reducing. (Sure, moving toward 8 will give better result if the end is in 8 and not in 8', otherwise the returning back to 7 is imminent.)

The adequacy between symbols for visual representation and representation that is necessary for decision making is obvious. No sophistication of the alternatives is required, however, descriptive and quantitative (statistical) generalizations are expected. Gradually increasing the complexity of the world, we expect to arrive at the self-generation of the major logical axioms, rules of inference, and even to such sophisticated concepts as strategies of search (and even A* algorithm).

10.5.7 A Possible Algorithm of Learning

One of the prerequisites for the learning is a capability of controlled system to sense the environment, and to drive to conclusions using the results of sensing. This input: sensing, and this output: conclusions the result of learning), are the input and output of an information processing system which is the object of our consideration.

Let us try to restore the stages of this information processing by analysis of these stages backward. The output of the learning system can be represented as a clause and ISA statement. A clause ("IF-THEN" statement) is a form in which any rule might be represented. All other statements of belonging to a known class, and the statement of forming a new class, can be unified as ISA statements. On the other hand ISA-statements is a particular case of the clause "IF ANYTHING, THEN AISA B".

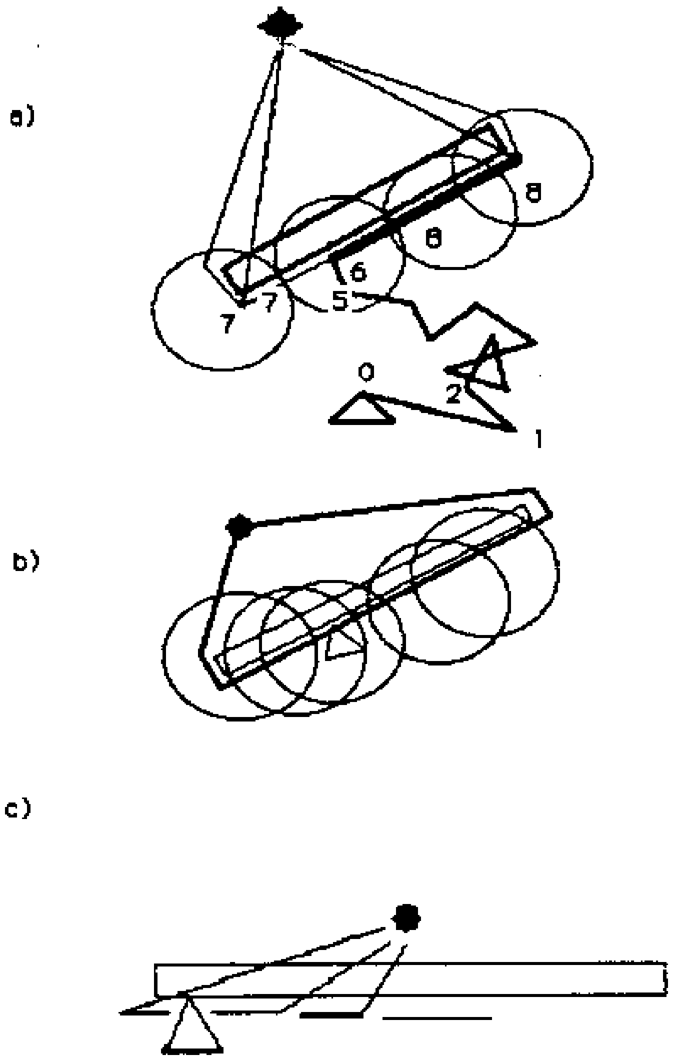


Figure 10-12 Baby-Robot in the World

Thus we will constrain our research by considering only clause statements, at the output. The following prerequisite for clause generation must be satisfied:

a) the words for antecedent, as well as for consequence, should already exist in the vocabulary of the system;

b) the implication link (IF-THEN) must be taught, or learned from an experience. For the stage of early cognitive development we do not assume any teaching. Thus the following questions must be answered:

1. How do the words appear in the vocabulary?
2. How is the implication link learned from an experience?

Labels could be assigned to any value of the sensor output, to any pixel of the CCD matrix, etc. Some measures should be undertaken for not labeling any distinguishable signal (at a given level of resolution). The only mechanism that seems to be applicable for this, in the possible repetitiveness of the values of the signal at the sensor output. The results with any type of similarity should be clustered on the basis of their *proximity* in space, in time, in magnitude, in sign, etc. So, the ability to make the comparison is presumed. The comparison determines difference among members of the similarity set. The following definitions can be introduced.

Definition 1. The set of similarity is a set in which the feature can be declared which is similar for all members of the set.

The set of similarity will be named a “class” after more than one set of similarity appears in the data base. In other words, similarity set is an initial array of non-classified signal. Classification procedure is applied to determine a similarity feature and a similarity set. The same procedure serves for distinguishing classes within similarity set.

Let us assume a set of objects with “A” property as a feature of similarity (A_1, A_2, ..., A_n) which is a similarity set by definition (other features are not known yet). The quantitative value v of A-property can be put in correspondence generating set of couples $\{(A_1\ v_1), (A_2\ v_2), ..., (A_n\ v_n)\}$. In fact, we are dealing with sequence of numbers given upon numerical axis which might be distributed randomly, uniformly have clots, etc.

A statement of existence can be formulated:

“(there exist)” cluster of A objects with a center at v_x.”

Relation between clusters can be found:

“(there exist)” a number of clusters with centers forming clusters...”

Behavior of these clusters may be analyzed in time.

An attempt to find implication can be made on the basis of experiences such as

if $t > 5$ A.M. then cluster at V_1

if $t > 11$ A.M. then cluster at V_2, etc.

In order to do this a mechanism of finding correlation should be devised which enables the system to notice the clot i happen to coincide with some period of time (comparison of moments of time is presumed.) Clearly, in order to make any implication, more than one modality of sensing should be considered (in this case sensing feature A and sensing time.)

10.5.8 Conditions of Clause Generation

Clauses can be formed when more than one modality of sensing is considered.

The following procedures must precede the appearance of the clause if the condition of clause generation is satisfied:

1) labeling the set by all of the existing sensing modalities;

2) putting in correspondence statement of existence of i-th modality with statement of existence of the j-th modality;

3) determining frequencies of these associations, and the relative frequencies;

4) linking the associated high frequencies "dots" as a clause;

5) putting the clause in long term memory.

Thus the existence of a short term memory, and a long term memory is implied. The short term memory is to deal with the auxiliary information at the stage preceding clause formulation; the long term memory should store the clauses. A question arises whether the short term memory should be cleared after the clause is formulated, or the material must be remembered for the feature associations. We will not discuss this question here since the Section is dedicated solely to the problem of early cognitive development.

The selection of a class with a double feature is equivalent to spotting the entity, or the object (examples: finding the edges on the basis of intensity and chain adjacency, or finding the blobs on the basis of intensity and spatial plane adjacency.)

So, the algorithm of early cognition can be presented as follows:

```
1.  RECOGNIZING FEATURES OF THE OBJECT FROM THE RESULTS OF SENSING
2.  LABELING FEATURES
3.  COMPARING OBJECTS BY LABELS
4.  GROUPING OBJECTS WITH SAME LABEL
5.  VALIDATING THE GROUPS [E.G. BY THE FREQUENCY OF APPEARANCE]
6.  IF MORE THAN ONE VALIDATION CRITERION THEN ASSIGN THE CLASS
7.  LABELING CLASS BY CLASS FEATURE
8.  COMPARING THE CLASSES BY CLASS FEATURES
9.  COMPARING THE OBJECTS WITHIN CLASS BY LABEL
10. GROUPING OBJECTS WITH SAME LABEL
11. VALIDATING THE GROUPS
12. IF MORE THAN VALIDATION CRITERION THEN ASSIGN THE SUB-CLASS
13. COUNT FREQUENCIES OF CLASSES
14. VALIDATE AN APPEARANCE OF FREQUENCIES
15. IF FREQUENCY IS GREATER THAN FREQUENCY THRESHOLD, AND
            IF THIS PHENOMENON APPEARS IN MORE THAN 1 CLASS
            THEN, PUT IN CORRESPONDENCE THE DISCOVERED FREQUENT
```

STATEMENTS AS A CLAUSE
16. COUNT FREQUENCIES OF SUBCLASSES........
and continue to position 14.

10.5.9 Learning when the Goal is Given

The process of cognition is described above is related to "unmotivated" cognition which is oriented toward proper organization of information from sensors. The efficiency of the information organization is determined only by criteria of memory space required, and time of storing and retrieval. The problem of adequacy of representation is not relevant since it presumes having the answer for a question "adequacy for what?" i.e. it implies existence of a goal.

In robots goal is strictly linked with mechanical motion. Let us assume without loss of generality that the goal is given in a form: to position the end-effector, or the mobile-system, in a definite point in a space and doing this with satisfaction of some goodness criterion (for example, minimum time, or minimum path.) This means that the features of class formation are already explicitly defined:

- relevance to the goal (measure of adjacency, closeness, etc.)
- relevance to the goodness criterion (e.g. minimum of time or distance).

This means that the labels which characterize the goal of the goodness criterion are the initial features of class formation, and the classes (and the clauses) which are formed the first are not built upon their independent frequency validation but rather use the goal and goodness labels as the first "centers of crystallization." On the other hand the consequent generation of subclasses can follow the procedure which has been already described above.

Thus the algorithm of early cognition for the goal oriented case can be written as follows:

- ASSIGN FEATURES TO THE OBJECT BY SENSING AND BY GOAL WORDS
- LABELING FEATURES
- VALIDATE OBJECTS BY GOODNESS CRITERION
- COMPARE OBJECTS BY LABELS
- COMPARE OBJECTS BY GOODNESS
- ASSIGN CLASS BY LABEL STATUS (A_{LS})
- ASSIGN SUBCLASS BY GOODNESS (A_G)
- STATE THE CLAUSE
  A_{LS} -> A_G OR A_G -> A_{LS}
- COUNT FREQUENCIES OF CLASSES
- VALIDATE APPEARANCE OF FREQUENCIES

…………………………………………………….

10.5.10 Simulation and Physical Experiments with Baby-Robot

A simulation experiment was developed to test Baby-Robot with the algorithm of generalization. The sensors return values of the distance to the target [D], the angle of heading [H], and the angle to the target [A]. The actuators are “go forward” [G] and rotate [R].

1. Goodness
2. Distance (D)
3. Heading (H)
4. Angle to Target (A)
5. Go forward (G)
6. Rotate(R)
7. D-H
8. D-A
9. D-G
10. D-R
11. H-A
12. H-G
13. H-R
14. A-G
15. A-R
16.G-R
17. D+H
18. D+A
19. D+G
20. D+R
21. H+A
22. H+G
23. H+R
24. A+G
25. A+R
26. G+R

Table 1: Enhanced Representation Sensors and Actuators in Baby Robot

The assigned goal is “Make D=0”. This goal will lead the learning sub-system to develop control rules that will allow Baby-Robot to achieve the goal. Following the algorithm of generalization, the algorithm searches the list of hypotheses and does not find any rule for performing the assignment. Thus it applies a random sequence to the actuators collecting experiences. The next step is the generalization algorithm. The first step is to separate the "good" experiences from the bad. In this case the goal is make D=O" the goodness of an experience is defined a "Delta D". The next step is to enhance the representation of these "good'" experiences. The enhanced representation is ordered as shown in Table 1. This notation will be used for all Figures [60-62].

Step by Step Recursive Generalization. The first step that the algorithm of generalization does is to check the database of Hypotheses. The only schema present in the database says that it should "Make D=O". Since there is nothing in the database about what to do in order to "Make D=O" it assigns "D=0" as the goal, and collects a random sequence of experiences. The next step in the algorithm of generalization is the creation of the classes of experiences. The clustering algorithm discovers two classes given the following rules of unification of classes:

If there are two consequent classes that have arbitrarily close maximums and the minimum separating them is arbitrarily close to the maximums, then the two classes are unified.

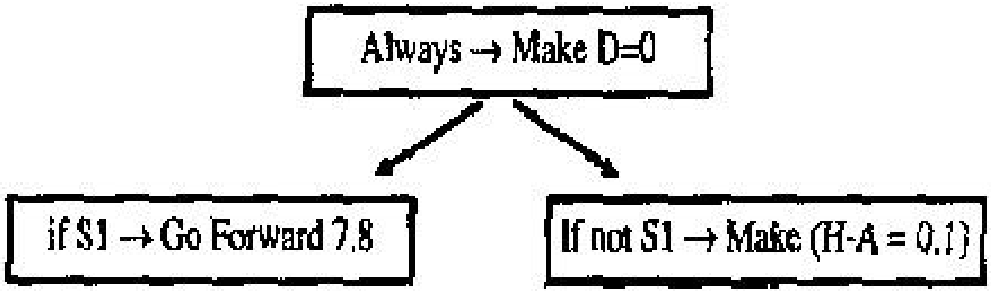


Figure 10-13: The database of Hypotheses at this step

If a class occupies the complete range in that coordinate, (i.e. there is only one class that includes all experiences in that coordinate) then the coordinate is discarded as rule candidate. They correspond to the actuator "go forward" and the enhanced representation sensor "Heading - Angle to Target". These two Hypotheses have an important difference. The first one gives a recommendation about what actuator to use in this situation. In the second one, the hypothesis says that if we are not in this situation we should go to this situation.

The problem with the second hypothesis is that there is nothing in the database of Hypotheses that gives instructions about how to "Make (H-A=0.1)". Thus, the generalization algorithm starts again:

1. all the experiences are re-ranked using the new goodness measure for the new goal;
2. the "best" experiences are selected using one of the previously selected methods;
3. these experiences are sent again to the classification algorithm, in other words, new eventgrams are created for the new goal and the classification algorithm creates new clusters which will form rules to follow the new goal.
4. a new goodness measure is taken by checking whether the experience brought it closer or further to the new goal;
5. the new goal is to "make H-A=0.1";

In Figure 10-14, we show the eventgrams for the same experiences and the new goal ("make H-A=0.1").

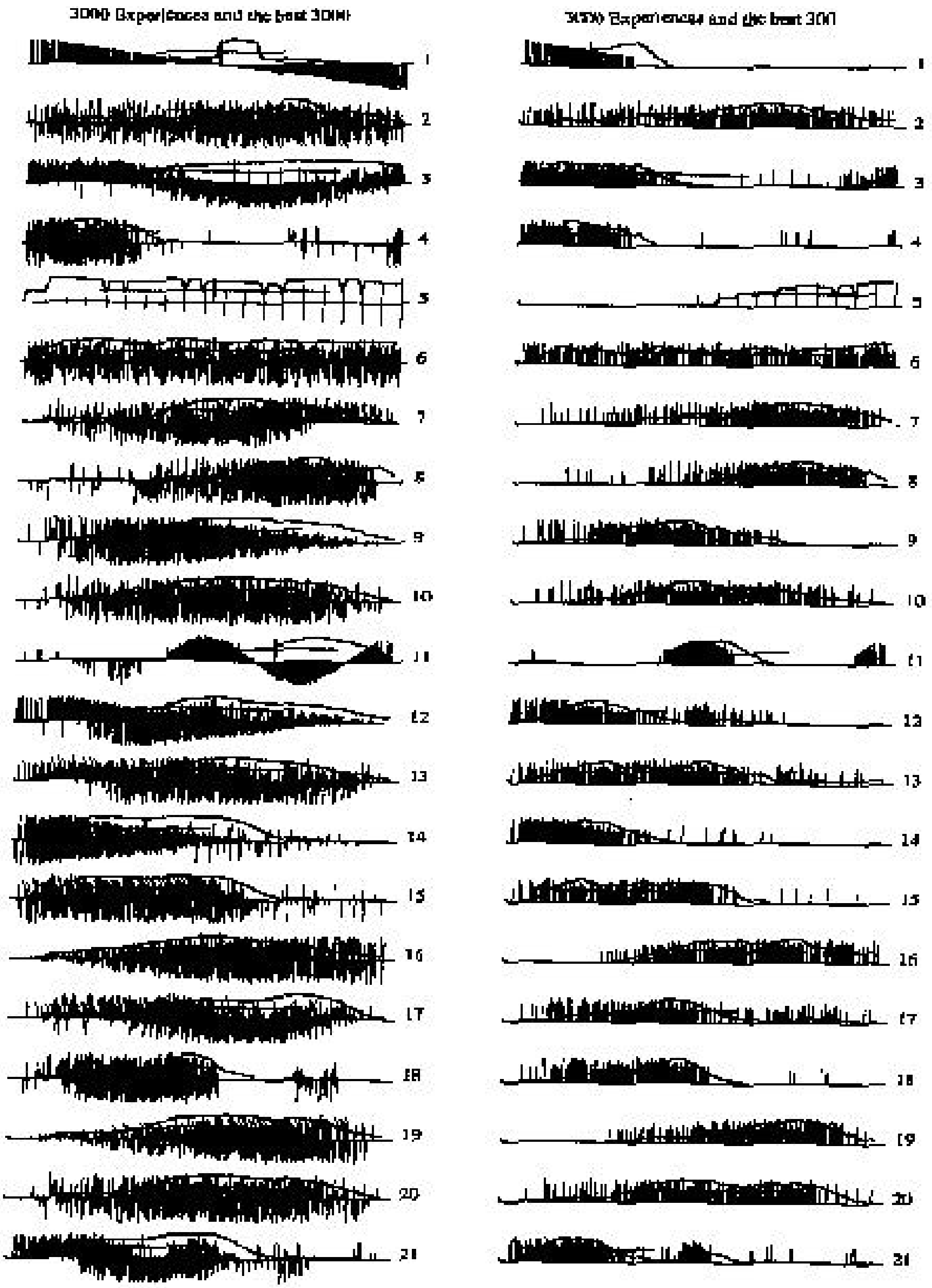
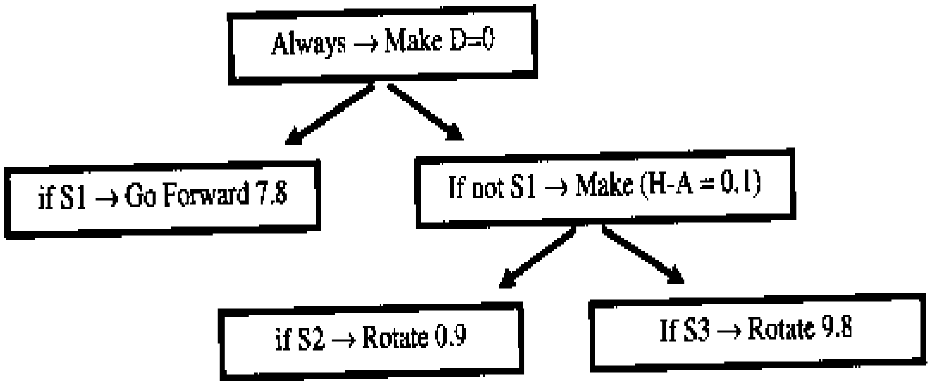


Figure 10-14 The eventgram

Figure 10-15 The new organization of knowledge



The experiences are sent to the averaging algorithm which finds two clusters in coordinate 6 which corresponds to the actuator "Rotate". These two clusters become two Hypotheses that get incorporated in the database as shown in Figure 10-15. In this figure "Rotate 0.9" and "Rotate 9.8" correspond to rotate right and rotate left. The "ST' is extracted from the experiences in which the rotate right command was applied and "SY from the rotate left experiences.

Baby Robot for 2D Operation

The robot for 2D Operation is called the Land Baby Robot (LBR). The simulation is shown in Figure 6, on the left we can see the mobile platform and on the right it is possible to see the target.

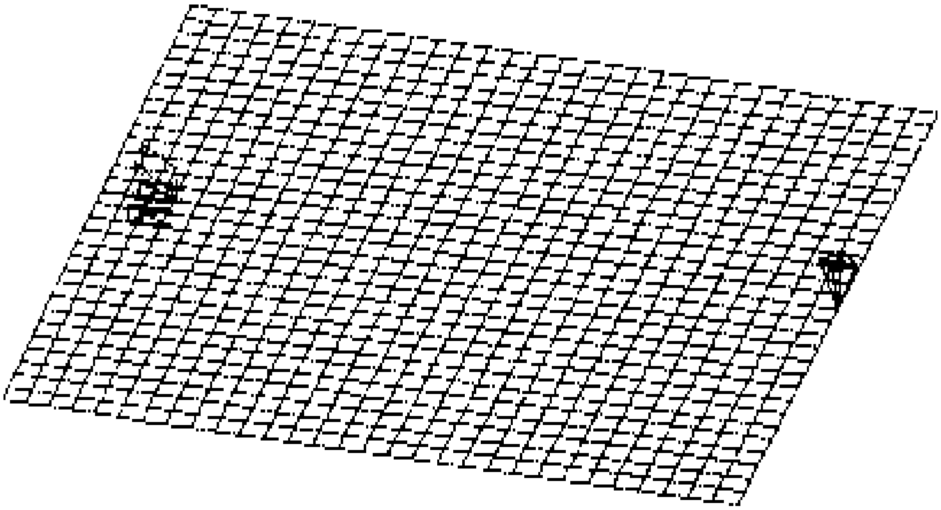


Figure 10-16 The situation at the beginning of simulation

Figure 10-17 shows a random path taken initially by LBR in order to collect experiences to create hypotheses and eventually schemata.

One can see that the trajectory of motion is very sophisticated one: it includes many possible single moves and their strings. Thus, not only the commands are verified, but also the ability to perform these commands. In other words, the strings of moves demonstrate the ability of our dynamic system to handle the input commands. The clauses inferred from the strings contain the dynamic model of the robot implicitly.

When the number of random moves is large one, it gives the richness of experiences which is satisfactory for inferring many possible rules. We have tested the quality of rules that can be inferred from the set of experiences of different volume. The results demonstrate that from the low total number of experiences, robot can infer the rule which has a large error. Increasing the total number of moves makes the rule more accurate.

If the system is capable of inferring from the changes in the rule that evolve when the number of test moves grows, then the system can notice the asymptotic character of these changes and the expected "correct" rule can be inferred before the total number of test moves approaches the infinity.

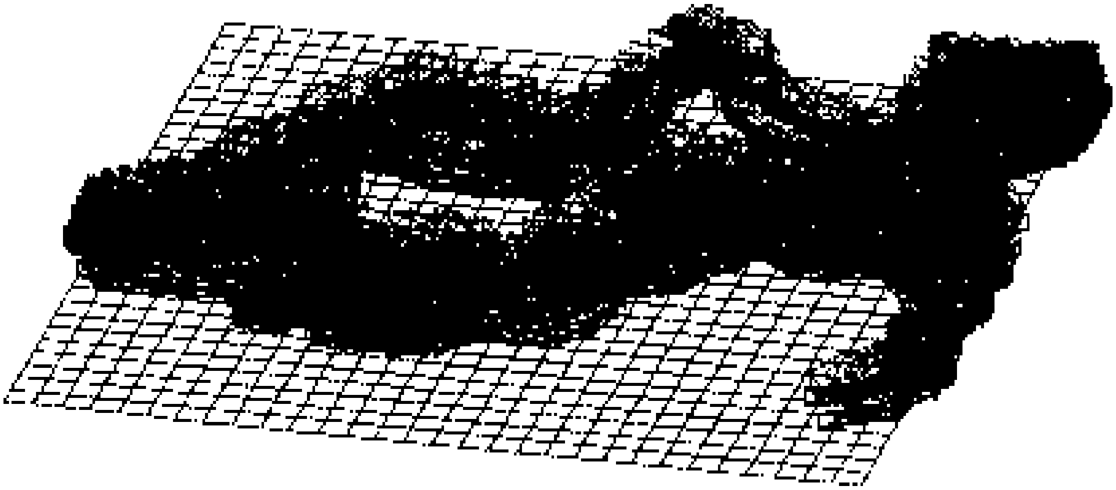


Figure 10-17 The trajectory of motion formed by a multiplicity of random moves

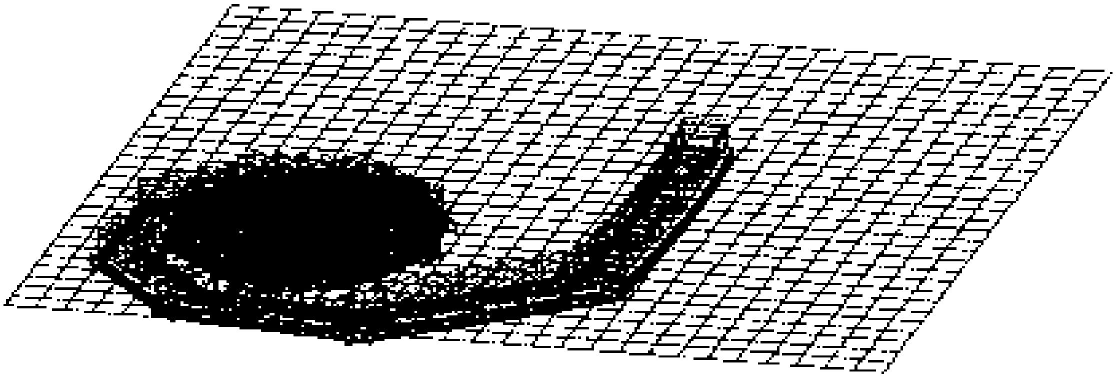
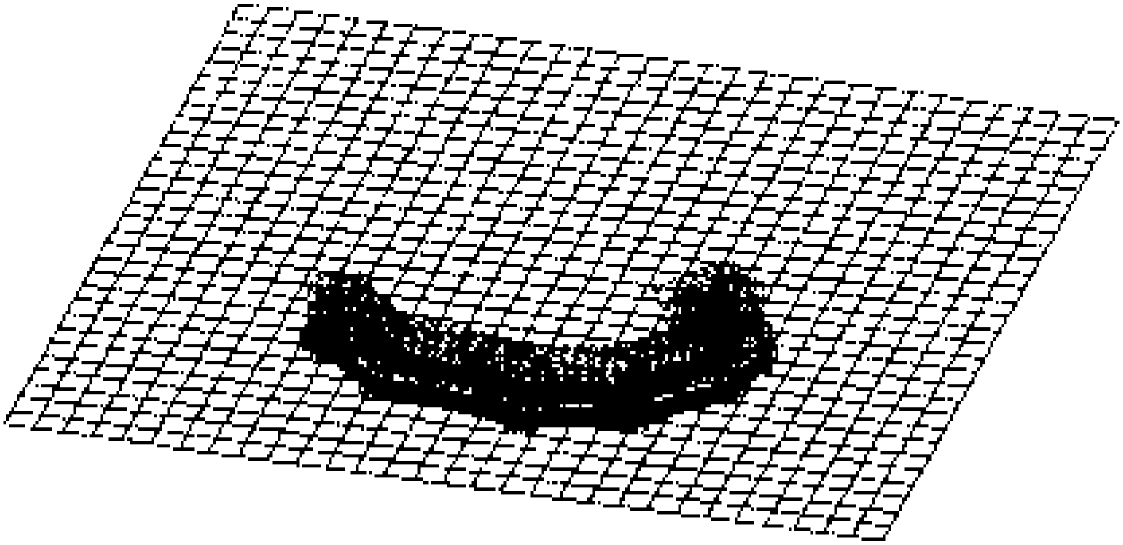


Figure 10-18. New trajectory: the first set of hypotheses is applied

The experiences are processed, and the first set of hypotheses allows for applying and verifying them. Figure 10-18 shows the behavior of LBR after the first set of hypotheses are collected. Certainly, the direction of the required motion is not computed with a sufficient accuracy. In the meantime, the new

Figure 10-19. Motion after the further improvements.



experiences of motion increase the database, and the new hypotheses arrive. The trajectory of motion becomes better. Figure 10-19 shows the numerical improvement upon the original hypothesis that yields better paths.

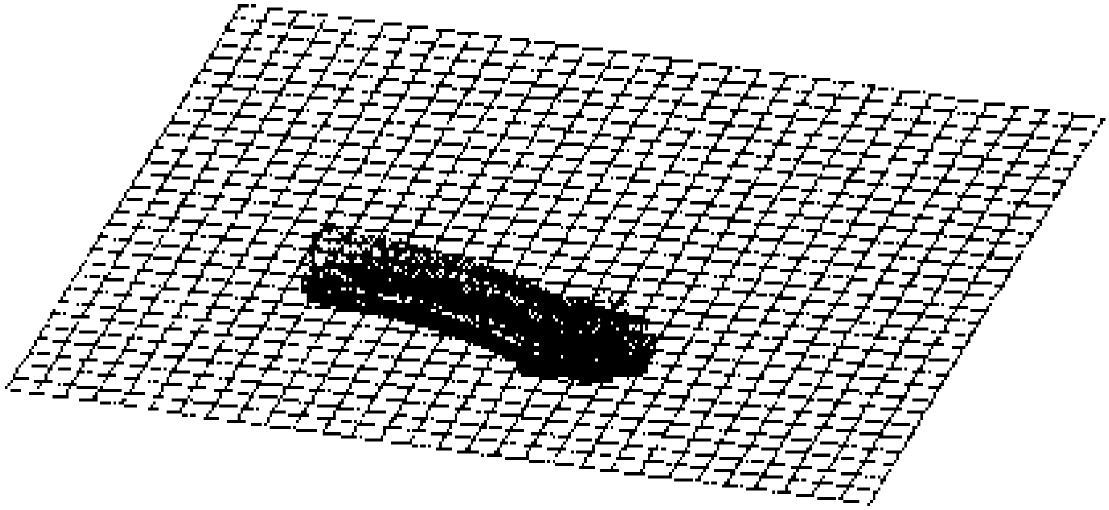


Figure 10-20 Quasi-optimum trajectory of motion

Figure 10-20 shows a quasi-optimal path that LBR converges to. It is possible to see that with the growth of the experience base, the error of the rule gets smaller .

Baby Robot for 3D Operation

The platform for 3D operation is called underwater Baby Robot (UBR). The sensors given to the UBR are:

1. Distance to the target.

2. The three Euler angles to the target.

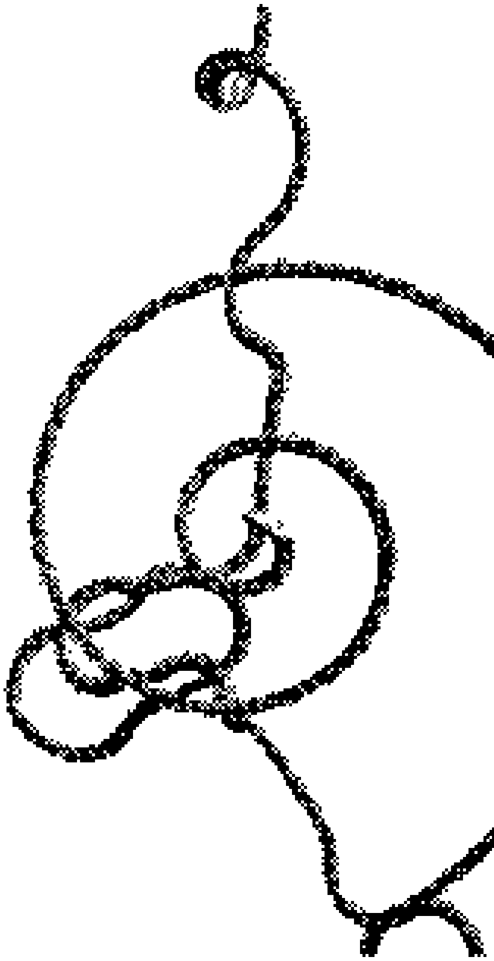


Figure 10-21 Database of hypotheses

And the actuators given to UBR are:

1. Propulsion.

2. Angles of the two rudders.

Figure 10-21 shows an example of a learned hypothesis. At the beginning, there are some random movements. Then, a hypothesis is generalized, showing a spiral movement of an incorrect hypothesis of spiraling. After more experiences are collected it will realize the performance could be improved and a new hypothesis will be generalized and this one may be thrown away.

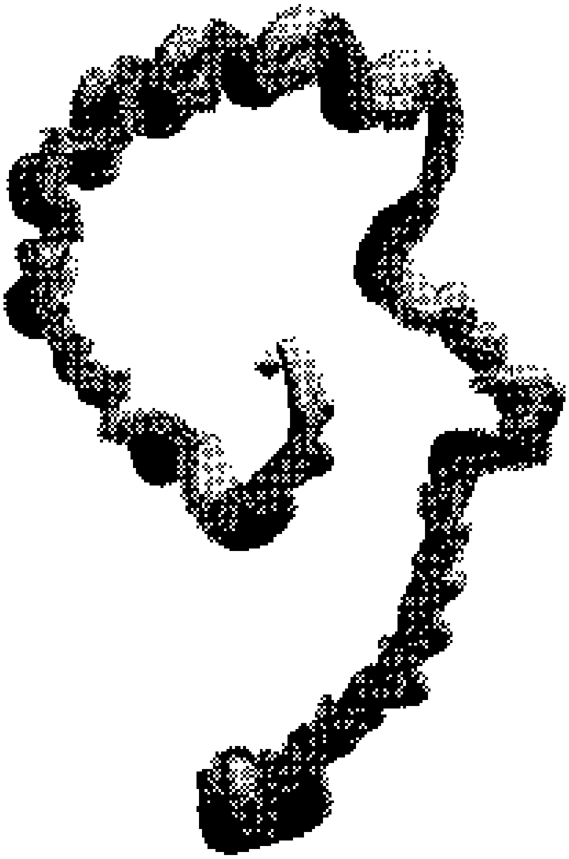


Figure 10-22 The first trial motion trajectory

Figure 10-22 shows the first trial to go to the goal. The traces were left on to show how complicated and superfluous is the path. Random movement and wrong hypothesis of spiraling control. More details about Baby-Sub and AstroBaby can be found in [60]. At the beginning of this trial the learning algorithm does not have any knowledge about the environment. It is possible to see four or five different hypotheses that create different motion.

Figure 10-23 shows some random movements and then the bang-bang control. Note, that we did not teach Our robot the concept of "bang-bang" control; it discovered it as a result of the learning process. After the first set of random movements we can see a sharp change in behavior. The first set of schemata is learned and Figure 10-23 shows that the submarine applies bang-bang

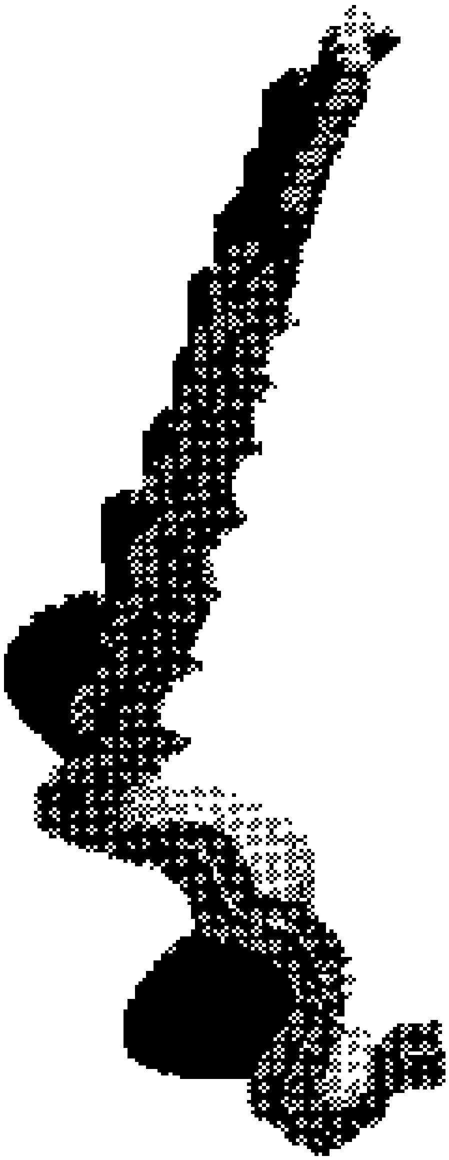


Figure 10-23: Bang-Bang control

strategy of control. Interestingly enough, "bang-bang" strategy was attributed to the maximum principle - a theoretically inspired method of optimization. The experience with Baby-Robot demonstrates that "bang-bang" can be discovered by the Baby-Robot at a very early stage of its learning.

10.5.11 Implications of Baby Robot Research

1. ECD processes in human and intelligent systems are compared and discussed as a key issue for determining the structures of learning processes. The need in bootstrap knowledge is determined, and its contents is delineated.

2. The difference between QL and CL is described. QL determines numerical values for the modules of the structures that are known prior to the process of learning while CL is associated with constructing new concepts as a result of generalization of previously existing concepts. The structure of CL is proposed for using in intelligent systems.

3. The concept of Baby-Robot is introduced as a tool for analysis of ECD processes in CL. It contains a semiosis mechanism: an elementary loop of functioning (ELF) containing sensors and actuators together with mechanisms of sensory processing and behavior generation connected to the storage of information. ELF equipped not only with an ability to store the experiences but also to organize and generalize them.

4. An algorithm that will serve as a subsystem for unsupervised learning is proposed for the Intelligent Controller Structure. It is based upon the algorithm of generalization and therefore contains operation of grouping, focusing attention and combinatorial search.

5. This system employs the method of nested clustering. The algorithm of generalization works recursively: it is applied to the result of its own functioning

6. The system simulates and executes the process of decision making working under condition of constantly growing set of rules. Thus, although the final goals of the system are known, the concrete way of actions that the system chooses might not necessarily be predictable by the external observer but it is always justified within the system of reasoning and the set of values submitted to IS at the stage of design.

7. The system organizes input information into a multiresolutional structure of world representation. The structure of knowledge base can be prepared for the consecutive growth of the hierarchy of representation.

8. Simulation experiments confirm the theoretical premises. The system was also manufactured as a hardware unit (Baby-Robot Senior Design Project at Drexel University).

References to Section 10.5

1. G.N. Saridis, "Intelligent Robotic Control", IEEE Transactions on Automatic Control, Vol. AC-28, No.5, 1983.

2. A. Meystel, "Intelligent Control of a Multiactuator System", In IFAC: Information Control Problems in Manufacturing Technology 1982, ed. D.E. Hardt, Pergamon Press, Oxford, 1983.

3. C. Isik, A. Meystel, "Knowledge -Based Pilot for an Intelligent Mobile Autonomous System", Proc. IEEE First Conference of AI Applications, Denver, CO. 1984.

4. G.R. Lefrancois, Psychology, Wadsworth Publ. Co., Belmont, CA. 1983.

5. J.A. Adams, Learning and Memory: An Introduction, The Dorsey Press, Homewood, IL. 1976.

6. C.P. Ward, Sensation and Perception, Academic Press, Orlando, FL. 1984.

7. J. MacLane, G. Birkhoff, Algebra, The MacMillan Col,New York, NY. 1967.

8. A. Hein, "The Development of Visually Guided Behavior", ed. C. Harris , in Visual Coding and Adaptability, Laurence Elbaum Associates, Hillsdale, NJ. 1980.

9. A.M.Farley,"Issues in Knowledge-Based Problems Solving", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-10, No.8, 1980.

10. E.B. Carne, Artificial Intelligence Techniques", Spartan Division, Washington DC, 1965.

11. R.L. Beurle, "Storage and Manipulation of Information in the Brain", Journal IEEE, Vol. 5, 1959.

12. R.L. Beurle, "Storage and Manipulation of Information in Random Networks", in Aspects of the Theory of Artificial Intelligence, ed. by C.A. Muses, Plenum Press, New York, NY, 1962.

13. M.D. Mesarovich, "On Self-Organizational Systems", in Self-Organizing Systems ed. by M.C. Yovitts, G.T. Jacobi, G.D. Goldstein, Spartan Books, Washington DC, 1962.

14. F. Rosenblatt, Principle of Neurodynamics, Spartan Books, New York, NY, 1962.

15 M. Minsky, S. Papert, Perceptions: An Introduction to Computational Geometry, The MIT Press, Cambridge, 1969.
16. B. Wizdow, M.E. Hoff, "Adaptive Switching Circuits", WESCON Conv. Record, Part IV, 1960.

17. S.-I, Amari, "Neural Theory of Association and Concept Formation", Biological cybernetics, Vol. 26, 1977.

18. T. J. Nelson, "A Neural Network Model for Cognitive Activity", Biological cybernetics, Vol. 49, 1981.

19. L. Bobrowsky, "Learning Processes in Multilayer Threshold Nets", Biological Cybernetics, Vol. 31, 1978.

20. L. Bobrowsky, E.R. Caiariello, “Comparison of Two Unsupervised Learning Algorithms”, Biological Cybernetics, Vol. 37, 1980.

21. A.D. Golt’sevy, E.M. Kussul, “Controlling the Motion of a Moving Robot Using a NN”, Automation and Remote Control, No 5, 1982.

22. D.F. Cuhn, S.R. Phillips, “ROBNAV: a Range-based Robot Navigation and Obstacle Avoidance Algorithm”, IEEE Transactions on Systems, Man and Cybernetics, No. 5, 1975.

23. A.G. Barto, R.S. Sutton, C.W. Anderson, “Neuron-like Adaptive Elements That Can Solve Difficult Learning Control Problems”, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-13, No. 5, 1983.

24 A.G. Barto, P. Anandan, “Pattern-Recognizing Stochastic Learning Automata”, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-15,No. 3, 1985.

25. A.G. Barto, C.W. Anderson, R.S. Sutton, “Synthesis of Nonlinear Control Surfaces by a Layered Associative Search Network”, Biological Cybernetics, Vol. 43, 1982.

26. A.G. Barto, R.S. Sutton, “Landmark Learning: An Illustration of Associative Search”, Biological Cybernetics, Vol. 42, 1981.

27. A.G. Barto, R.S. Sutton, P.S. Browser, “Associative Search Network: A Reinforcement Learning Associative Memory”, ”, Biological Cybernetics, Vol. 40, 1981.

28 K. Fukushima, S. Miyake, T. Ito, “New Cognition: A Neural Network Model for a Mechanism of Visual Pattern Recognition”, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-13,No. 3, 1983.

29. R.R. Gawzonski, B. Macukow, “The Adaptive Neuron Like Layer Net as a Control Learning System Research”, Vol. III, Adv. Publ. Ltd. London, 1977.

30. R.R. Gawzonski, “Two Stage Learning Algorithm for Optimal Control and a Manipulator”, Proc. of the Conference on Robotic Research: The Next Five Years and Beyond, Bethlehem, PA. 1984.

31. J.J. Craig “Adaptive Control of Manipulators Through Repeated Trials”, Proc. of the 1984 ACC, Vol. 3. San-Diego, CA, 1984.

32. D. Stove, “Hume, Probability, and Induction, Philosophical Review, April, 1965.

33. G. Cohen, P. Bernhard, “On the Rationality of Some Decision Rules in a Stochastic Environment”, IEEE Transactions on Automatic Control, Vol. AC-24, No. 5, 1979.

34. S.R. Liberty, R.C. Hartwig, “Design-performance-measure statistics for stochastics linear control systems”, IEEE Transactions on Automatic Control, Vol., AC-23, 1978.

35 S.M. Brooks, “A Comparison of maximum seeking methods”, Operations Research, V. 7, No. 4, 1959.

36. A.N. Mucciardi, “Self-Organizing Probability State Variable Parameter Search Algorithms for Systems that Must Avoid High-Penalty Operation Regions”, IEEE Transactions on Systems, Man and Cybernetics, Vol. SM4-4, No. 4, 1974.

37. S. Hampson, D. Kibler, “A Boolean Complete Neural Model of Adaptive Behavior”, Biological Cybernetics, Vol. 49, 1983.

38. J. Simons, et al., “A Self-Learning Automation with Variable Resolution for High Precision Assembly by Industrial Robots”, IEEE Transactions on Automatic Control, Vol. AC-27, No. 4, 1982.

39. L. Siklossy, J. Dreussi, “An Efficient Robot Planner Which Generates Its Own Procedures”, in Robotic Research,

Proc. of the 3-d IJCAI, Stanford, CA, 1973.

40. B. Dufay, J.C. Lalombe, “An Approach to Automated Robot Programming Based on Inductive Learning”, in Robotic Research, The First Intel. Symp., ed. by M. Brady and R. Paul, MIT Press, 1984.

41. R. E. Fikes, P.E. Hart, N.J. Nilsson, “Learning and Executing Generalized Robot Plans”, Artificial Intelligence, 3, 1972.

42. P.H. Winston, “Learning Structural Description from Examples”, in The Psychology of Computer Vision, NY, McGraw Hill, 1975.

43. P.H. Winston, T.O. Binford, B. Katz. M. Lowry, “Learning Physical Descriptions, Examples and Precedents”, in Robotic Research, The First Intel. Symp., ed. by M. Brady and R. Paul, MIT Press, 1984.

44. W.S. Havens, “Recognition Mechanisms for Schema-Based Knowledge Representations”, Comp. & Maths. with Appls. Vol. 9, No. 1, 1983.

45. J.M. Tenenbaum, H.G. Barrow, “IGS: A Paradigm for Integrating Image Segmentation and Artificial Intelligence, ed. by C.C. Chen, Academic Press, NY, 1976.

46. M.D. Levene, “A Knowledge-Based Computer Vision System”, In Computer Vision Systems, ed. by A.R. Hanson and E.M. Riseman, Academic Press, NY, 1978.

47. A.R. Hanson, E.M. Riseman, “A Design of Semantically Directed Vision Processor”, Comp. and Info. Sci. Dept. Tech, Rep. 75C-1 Univ. of Mass.

48. R. Bajcay, “Visual and Conceptual Focus of Attention”,In Structured Computer Vision Systems, Academic Press, NY, 1980.

49. T. Lozano-Perez, “A Language for Automatic Mechanical Assembly”, in Artificial Intelligence, an MIT Perspective, V. 2, ed. by P.H. Winston and R.H. Brown, The MIT Press, 1979.

50. H. Farreny, H. Prade”On the Problem of Identifying an Object in a Robotic Scene from a Verbal Imprecise Description”, in Advanced Software in Robotics, ed. by A. Danthine and M. Geraldin, North-Holalnd, 1984.

51. H. Jappinen, “A Model for Representing and Invoking Robot Bodily Skills”{, in Advanced Software in Robotics, ed. by A. Danthine and M. Geraldin, Elsevier, North-Holalnd, 1984.

52. J.H. Graham, G.N. Saridis, “Linguistic Decision Structures for Hierarchical Systems”, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-12, No.3, 1982.

53. G.N. Saridis, J.H. Graham, “Linguistic Decision Schemata for Intelligent Robots”, Automatica, Vol. 20, No. 1, 1984.

54. Tu-Shing Chen, “Image Processing by Incomplete Information”, IEEE Workshop on Languages of Automation, Mallorca, Spain, 1985.

55. J. Albus. Outline for a theory of intelligence. *IEEE Transactions on Systems, Man, and Cybernetics,* 21, 1991.

56. A. Meystel. Multiresolutional architecture for autonomous systems with incomplete and inadequade knowledge representation. H. B. Tzafestas and H.B. Verbruggen, editors, *Al in Industrial Decision Making, Control and Automation,* 1995.

57. A Meystel. Baby-robot: on the analysis of cognitive controllers for robotics. *Proc. IEEE Int'l Conf. on Systems, Man, and Cybernetics,* 1985.

58. Eilbert, A. Meystel, L. Venetsky, and S. Ziets. Baby robot: Learning to control goal-oriented behavior. *Proc. IEEE Workshop on Intelligent Control,* 1985.

59. Lacaze, M. Meystel, and A. Meystel. Schemata for unsupervised learning fo autonomous robots for 3d space operation. *NASA Conference on Space Applications of Al,* pages 103-112, Greenbelt, MD, 1993.

60. Lacaze. Unsupervised early conceptual learning in mobile robots. Master's thesis, Drexel University, 1994.

61. Y. Maximov and A. Meystel. Optimum design of multiresolutional hierarchical control systems. *Proceedings of IEEE Int'l Symposium on Intelligent Control,* pages 514-520, Glasgow, U.K., 1992.

J.K. Feibleman. *An introduction to the Philosofy of Charles S. Peirce.* MIT Press, 1946.

10.8 Applying Neural Networks for Learning

From the five previous Sections we have established a vision that Learning is a Generalization upon Multiple Information. This multiple information can be collected within a temporal window around the time instant, and it can be collected within a spatial window around a particular instant of space. The process of generalization is understood as discovering entities of lower resolution out of information initially presented in a much higher resolution. It is done usually by some kind of probabilistic averaging (determining the “moments”), and the computation can be done by special architectural modules: neural networks which are specialized multiprocessors with all advantages of parallel processing versus single-processor computation.

10.8.1 Neural Networks: Architectures for Generalization of Multiple Information

Neural Networks are conceptual architectures for dealing with multiple information. Each architecture is constructed from *modules,* designed to solve parts of a bigger problem of generalization. It allows to substitute representation using many samples of high resolution by another representation of lower resolution. Since the processing can be performed on a single processor computer as well as with multiprocessor module, we would stress the conceptual importance of Neural Networks as a mechanism of generalization. Thus, we can consider Neural Nets as both the metaphorical and hardware way of dealing with multiple information.

Fundamental standard solutions are developed in the areas of multiple information processing modules (NN-modules) for

- associative memory,
- pattern recognition, and

- category learning.

Various ideas are explored in the process of developing of these solutions. The following milestones in NN development are considered:

- McCulloch-Pitts neuron,
- perceptrons,
- adaline and madaline,
- back propagation,
- the learning matrix,
- linear associative memory,
- embedding fields,
- instars and outstars,
- competitive learning,
- adaptive resonance theory ,
- the cognitron and neocognitron,

McCullochPitts Neuron. The McCullochPitts model [1] describes a neuron whose output is the sum of inputs that arrive via weighted pathways. The input from a particular pathway is an incoming signal multiplied by the weight of that pathway. These weighted independent inputs are summed. By assigning weights, one can obtain the weighted average for any probabilistic law. The outgoing signal is typically a nonlinear function-binary, sigmoid, threshold or linearly growing input signal in that cell. The McCulloch-Pitts neuron can also have a bias term, which is formally equivalent to the negative of a threshold of the outgoing signal function.

Actually, McCulloch-Pitts neuron makes grouping (by weighted summation) and focusing attention (by fanning) but not combinatorial search. A convenient notation is known for describing the McCulloch-Pitts neuron, called the *adaptive* filter [2]. The elementary adaptive filter has the following components:

- a level that registers an input pattern vector; it performs a function of focusing attention through “fanning” the signals that pass through weighted pathways; i.e. focusing attention is performed and

- a level which summarizes these inputs, i.e. grouping is performed.

This principle of operation has proved useful because the adaptive filter computes a pattern match, by producing the output function that can be a binary step (a decision), a linear growth, or a sigmoid signal. In 1943, McCulloch and Pitts analyze this device *without adaptation* i.e. without adjustment of the weights [2]. This adjustment (adaptation) can be considered a search for the *proper* grouping, and thus is an elementary learning.

Perceptron: Learning with Feedback Compensation. Next generation of researchers started exploring McCulloch-Pitts neuron for repetitive learning and adaptation. Perceptron was developed by Rosenblatt in 1958 [3]. The core idea of the perceptron is a development of rudimentary learning by assigning weights. In the Perceptron matrix, the results of collective learning by the multiple neurons are focused upon, grouped as the results allow for, and searched again for matching within them. The sub-modules of Perceptron were called: the sensory unit; the association unit, and the response unit.

One of the many perceptrons that Rosenblatt studied, was the *back-coupled perceptron* [4]. This model is equipped with a feedforward adaptive filter and has a binary output signal. The weights fanning in to the particular node are adjusted in proportion to the error at that node. The actual output vector is subtracted from the target output vector; their difference is defined as the error; and that difference is then fed back to adjust the weights, according to some probabilistic law (back-coupled error correction). This model of two-level perceptron could sort linearly separable inputs into two classes.

Minimizing the Group Value of Error. The new set of perceptron-models used was created by B. Widrow and his colleagues: *adaline and madaline* perceptrons. The adaline model has just one neuron at the intermediate level, while the madaline, or many-adaline, model has any number of neurons in that level. An adaline/madaline model compares the *analog* output x, with the target output b. This comparison provides a more subtle judgment of error than a law that compares the *binary* output with the target output [5, 6]. The error was fed back to adjust weights using a Rosenblatt back-coupled error correction rule. Their system is sometimes referred to as *least mean squared* error correction rule, or LMS.

Multilevel Perceptrons and Back-propagation. Very soon, the multiple-level perceptrons emerged. First, they were introduced in 1962 by Rosenblatt.. He has introduced the term “back-propagation” in the Section called "Back-propagating error correction procedures” of his book [4]. His back-propagation algorithm used a probabilistic learning law. Present versions of back-propagation combine the McCulloch-Pitts linear filter with a sigmoid output signal function and with Rosenblatt back-coupled error correction. The modern version was created by P. Werbos in 1974 [7] (and independently developed by D. Parker in 1982 [8]). A very frequent scheme of back-propagation performs the associative learning: during training: one vector pattern is associated with another. After completion of training, the second pattern is recalled in response to input of the first [9]. The back propagation system is trained under conditions of *slow learning,* with each pattern presented repeatedly during training (at this stage, the combinatorial search is performed). In a multilevel perceptron, the input signal vector converges on the "hidden unit" level after passing through the first set of weighted pathways. Fanning between levels plays the role of “focusing attention”. Output signals fan out to the level, which generates the actual output of this feedforward system (the results of grouping). A back-coupled error correction system compares the actual output with a target output and feeds back their difference to all the weights.

Learning Matrix. In the aftermath of perceptron, many of the models followed the “Hebbian rule of learning” determining a correlation between the presynaptic and the postsynaptic signals. This rule provides for a qualitative description of increases in path strength that occur when one cell helps to fire another. In the adaptive filter formalism, this hypothesis is often interpreted as a weight change that occurs when a presynaptic signal is correlated with a postsynaptic activity [10]. One of the earliest NN modules based upon Hebbial rule is the learning matrix developed by K. Steinbuch [11]. The function of the learning matrix is to sort, or partition, a set of vector patterns into categories (classification). This model is the precursor of a fundamental module widely used in present day neural network modeling and called *competitive learning*. Steinbuch's learning rule can be translated into the Hebbian formalism, with weight adjustment during learning a joint function of a presynaptic and postsynaptic signals. A comparative analysis of the learning matrix, the madaline models, and their electronic implementations can be found in [12]. Many developments incorporate Linear Matrix and Hebbian Rule of Learning. It is related to Linear Associative Memories developed and described in [13-17].

Real-time modeling. The desire to avoid the external control of NN-modules led to the theory of *embedding fields* [18] which has allowed to combine the *fast* nodal activation and the *slow* weight adaptation. Two key architectural components of embedding field systems are the *instar* and *outstar* units. Instars appear in systems designed to carry out adaptive coding, or content-addressable memory (CAM). The outstar, which is dual to the instar, carries out spatial pattern learning. Powerful computational properties arise when neural network architectures are constructed from a combination of instars and outstars. The outstar and the instar have been studied in great detail and with various combinations of activation, or short-term, or long-term memory and equations. A review of neural models that are versions of the additive equation can be found in [19]. A series of theorems encompassing neural network pattern learning by systems employing a large class of these and other activation and learning laws (including *outstar learning theorems*) was proved in [20, 21]. In general, the order of activation of the outstars, as well as the spatial patterns themselves, need to be learned. This can be accomplished by using auto-associative networks, as in the theory of serial learning [22, 23]

Competitive Learning. A module for *competitive learning* brings the properties of the learning matrix into the real-time setting. The basic competitive learning module combines the instar pattern coding system with a competitive network that contrast and enhances its filtered input. The basic competitive learning architecture consists of an instar filter, and a competitive neural network. The competitive learning module can operate with or without an external teaching signal; and learned changes in the adaptive filter can proceed indefinitely or cease after a finite time interval. If there is no teaching signal at a given time, then the net input vector is the sum of signals arriving via the adaptive filter. Then if the category representation network is designed to make a choice, the node will automatically becomes active if its weight vector matches the signal vector the best. If there is a teaching signal, the category representation still depends on past learning, but this is balanced against the external signal, which may or may not overrule the past in the competition. In either case, an instar learning law allows a chosen category to encode pattern in its learned representation. Systems were designed to learn computational maps, producing an output vector in response to an input vector. In the core of many of these computational map models there is an instar-outstar combination [24-31]. The self-organizing feature map [16] and the counter-propagation network [32] are also examples of instar-outstar competitive learning models.

Adaptive Resonance Theory. Analysis of the instability cases for feedforward instar-outstar systems led to the introduction of adaptive resonance theory (ART) [26] and to the development of neural network systems ART I and ART 2 [25, 26]. ART networks are designed, in particular, to resolve the stability-plasticity dilemma: they are stable enough to preserve significant past learning, but nevertheless remain adaptable enough to incorporate new information when it appears. The minimal ART module includes a bottom-up competitive learning system combined with a top-down outstar pattern learning system. When an input is presented to an ART network, the system dynamics initially follows the course of competitive learning with bottom-up activation leading to a category representation with enhanced contrast. In the absence of other inputs, the active category is determined by the past learning as encoded in the adaptive weights in the bottom up filter. But now, in contrast with feedforward systems, signals are sent via a top-down adaptive filter. This feedback process allows the ART module to overcome all of the sources of instability. A minimal ART module is a category learning system that self-organizes a sequence of input patterns into various recognition categories. It is not an associative memory system. However, like the competitive learning module in the 1970s, a minimal ART module can be embedded in a larger system for associative memory. ART systems can also be used to pair sequences of the *categories* self-organized by the input sequences. The symmetry of the architecture implies that pattern recall can occur in either direction during the performance. This scheme brings to the associative memory paradigm the code compression capabilities of the ART system, as well as its stability properties [32-34].

Cognitron and Neocognitron. conclusion, we will consider two sets of models that are variations on the themes previously described. The first class, consists of the cognitron and the larger-scale neocognitron [35-37]. This class of neural models is distinguished by its capacity to carry out translation-invariant and size-invariant pattern recognition. This is accomplished by redundantly coding elementary features in various positions at one level; then cascading groups of features to the next level; then groups of these groups; and so on. Learning can proceed with or without a teacher.

Locally the computations are a type of competitive learning that use combinations of additive and shunting dynamics.

* *

*

Most of the above developments contributed into CMAC: Cerebellar Model Arithmetic Controller[53]. CMAC has the following properties: local generalization, rapid algorithmic computation based on LMS training, incremental training, functional representation, output superposition, and a fast practical hardware realization. Next Section is dedicated to the explanation of how CMAC works is provided, and to descriptions of CMAC applications in robot control, pattern recognition, and signal processing.

REFERENCES

1. Carpenter, G. A. "Neural Network Models for Pattern Recognition and Associative Memory", *Neural Networks, vol. 2,* 1989

2. McCulloch, W. S., & Pitts, W. (1943). “A logical calculus of the ideas immanent in nervous activity”. *Bulletin of MathematicalBiophysics, 9,* pp.127-147

3. Rosenblatt, F., “The perceptron: A probabilistic model for information storage and organization in the brain”. *Psychological Review, 65,* 1958, pp. 386-408

4. Rosenblatt, R., *Principles of neurodynamics.* Washington, DC: Spartan Books, 1962

5. Widrow, B., & Hoff, M. E. Adaptive switching circuits. *1960 IRE WESCON Convention Record,* part 4, 1960, pp. 96-104

6. Widrow, B., & Winter, R. “Neural nets for adaptive filtering and adaptive pattern recognition”. *Computer, 21*, 1988, pp. 25-39

7. Werbos, P. J. *Beyond regression: New tools for prediction and analysis in the behavioral sciences.* Unpublished Ph.D. thesis, Harvard University, Cambridge, MA, 1974

8. Werbos, P. J. “Generalization of backpropagation with application to a recurrent gas market model”. *Neural Networks*, 1, 1988, pp. 339-356

[53] CMAC was originally interpreted as Cerebellar Model Articulation Controller: it was invented for a pratical application in the systems of control of multilink manipulators where the problem of motion articulation has the highest priority. Later, it became clear that CMAC has broader application, and it was renamed with focusing upon its general computational capabilities.

9. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. “Learning internal representations by error propagation”. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructures of cognitions, v. I,* Cambridge, MA: MIT Press, 1986, pp. 318-362

10. Grossberg, S. “Some nonlinear networks capable of learning a spatial pattern of arbitrary complexity”. *Proceedings of the National Academy of Sciences USA, 59,* 1968, pp. 368-372

11. Steinbuch, K. “Die Lernmatrix”. *Kybernetik, 1,* 1961, pp. 36-45

12. Steinbuch, K., & Widrow, B. “A critical comparison of two kinds of adaptive classification networks”. *IEEE Transactions on Electronic Computers,* **EC-14,** 1965, pp. 737-740

13. Kohonen, T. “Correlation matrix memories”. *IEEE Transactions on Computers, C-21,* 1972, pp. 353-359

14. Anderson, J. A. “A simple neural network generating an interactive memory”. *Mathematical Biosciences, 14,* 1972, 197-220

15. Kohonen, T. *Content-addressable memories.* Berlin: Springer-Verlag, 1980

16. Kohonen, T. *Self-organization and associative memory.* Berlin: Springer-Verlag, 1984

17. Nakano, N. “Associatron: A model of associative memory”. *IEEE Transactions on Systems, Man, and Cybernetics, SMC2,* 1972, 381-388

18. Grossberg, S. *The theory of embedding fields with applications to psychology and neurophysiology.* New York: Rockefeller Institute for Medical Research, 1964

19. Grossberg. S. “Nonlinear neural networks: Principles, mechanisms, and architectures”. *Neural Networks,* 1, 1988, pp. 17-61

20. Grossberg, S. “Pattern learning by functional-differential neural networks with arbitrary path weights”. In K. Schmitt (Ed.), *Delay and functional differential equations and their applications (pp.* 121-160). New York: Academic Press, 1972

21. Grossberg, S. “Neural expectation: Cerebellar and retinal analogs of cells fired by learnable or unlearned pattern classes”. *Kybernetik, 10,* 1972, pp. 49-57

22. Grossberg, S., & Pepe, J. “Schizophrenia: Possible dependence of associational span, bowing, and primacy vs. recency on spiking threshold”. *Behavioral Science, 15,* 1970, pp. 59-362

23. Hecht-Nielsen, R. “Neural analog information processing”. *Proceedings of the Society of Photo-Optical Instrumentation Engineers,* ***298,*** 1981, pp. 138-141

24. Grossberg, S. “On the development of feature detectors in the visual cortex with applications to learning and reaction diffusion systems". *Biological Cybernetics, 21,* 1976, pp. 145-159

25. Grossberg, S. “Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors”. *Biological Cybernetics, 23,* 1976, pp. 121-134

26. Grossberg, S. “Adaptive pattern classification and universal recoding II: Feedback, expectation, olfaction, and illusions”. *Biological Cybernetics, 23,* 1976, pp. 87-202

27. Amarl, S.-I. “Neural theory of association and concept formation". *Biological Cybernetics, 26,* 1977, pp. 175-185

28. Amari, S.-I., & Takeuchi, A. “Mathematical theory on formation of category detecting nerve cells”. *Biological Cybernetics,* 29, 1978, pp. 127-138

29. Bienenstock, E., Cooper, L. N., & Munro, P. W. “A theory for the development of neuron selectivity: Orientation specificity and binocular interaction in the visual cortex". *Journal of Neuroscience, 2,* 1982, pp. 32-48

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

31. Rumelhart, D. E., and Zipser, D. “Feature discovery by competitive learning”. *Cognitive Science, 9,* 1985, pp. 75-112

32. Hecht-Nielsen. R. “Counterpropagation networks”. *AppliedOptics, 26,* 1987, pp. 4979-4984

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

34. Carpenter, G. A., & Grossberg, S. “ART 2: Self-organization of stable category recognition codes for analog input patterns”. *Applied Optics, 26,* 1987, pp. 4919-4930

35. Fukushima, K. “Cognitron: A self-organizing multilayered neural network”. *Biological Cybernetics,* 20, 1975, pp. 121-136

36. Fukushima, K. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". *Biological Cybernetics,* 36, 1980, pp. 193-202

37. Fukushima, K. “Neocognitron: A hierarchical neural net work capable of visual pattern recognition”. *Neural Networks,* 1, 1988, pp. 119-130

10.8.2 CMAC: An Associative Neural Network Alternative to Backpropagation

Learning with CMAC. In this sub-section we describe a neural network called CMAC (Cerebellar Model Arithmetic Controller[54]. CMAC has been designed for real-time control of an

[54] It was called originally "Cerebellar Model Articulation Controller" since it was designed for the needs of intelligent robots. Later, it became clear that its domain of application is much broader, and CMAC acquired more general - and more accurate - name: "Cerebellar Model Arithmetic Controller"

industrial robot and in other applications, typically employing neural networks. CMAC has allowed to represent and use hundreds of thousands of adjustable weights that can be trained to approximate nonlinearities which are not explicitly written out or even known. CMAC can learn relationships from a very broad category of functions. Furthermore, the learning algorithm generally converges in a small number of iterations.

Neural networks of one form or another have been "modeled" for a number of decades. Well known models include the Rosenblatt's Perceptron [3], the Widrow's Adaline [4, 5], and Hopfield's recurrent networks [6, 7]. Great attention was focused on the problem of training multilayer NNs. Backpropagation has been developed as a method of training multilayer networks, and most recent application papers use this approach [43]. The CMAC neural network is an alternative to the backpropagation-trained analog multilayer neural network. An alternative is useful since backpropagation has a number of disadvantages. It: requires many iterations to converge and therefore is inappropriate for online real-time learning. It produces a large number of computations per iteration so that the algorithm runs slowly unless implemented in expensive custom hardware. It has an error surface with possible local minima hazardous to backpropagation training that is based on gradient search techniques. It does not allow for productive incremental learning: all inputs must be seen before any weight change can take place if convergence should be quickly achieved.

In the 1970s J. Albus reported the work on CMAC [1, 2, 8, 9]. CMAC was applied for rote learning of movements of an artificial arm. After a number of years the practicality of CMAC can be considered unquestionable. Many additional advantages were discovered. For example, CMAC can be used to learn general state space dependent control responses [10]. The Robotics Laboratory at the University of New Hampshire has been investigating and using CMAC with considerable success [10-22]. Other groups working with CMAC include Ersu et al. [23-26], and Moody at Yale [27]. Several industrial research groups using CMAC. CMAC became a key neural module for a multiplicity of works in machine learning and neuro-control.

CMAC is an associative neural network. Only a small subset of the network influences any instantaneous output, and that subset is determined by the input to the network. The associative mapping built into CMAC assures local generalization: similar inputs produce similar outputs while distant inputs produce nearly independent outputs. As the result of the built-in associative properties, the number of training passes required for network convergence on real problems is orders of magnitude smaller with CMAC than with backpropagation.

Architecture of CMAC. In this sub-section we describe CMAC as an automaton: the input space, the outputs, its inner functions, and the mechanism of generalization. Figure 10-21 demonstrates an overview of CMAC [29]. Its input, state space and the conceptual memory A are N-dimensional. The actual memory A' has as many dimensions as there are output components. An input vector is the collection of N appropriate sensors of the real world and/or measures of the desired goal. The input space consists of the set of all possible input vectors. The number N of input vector components and the number of outputs is arbitrary within some practical limits.

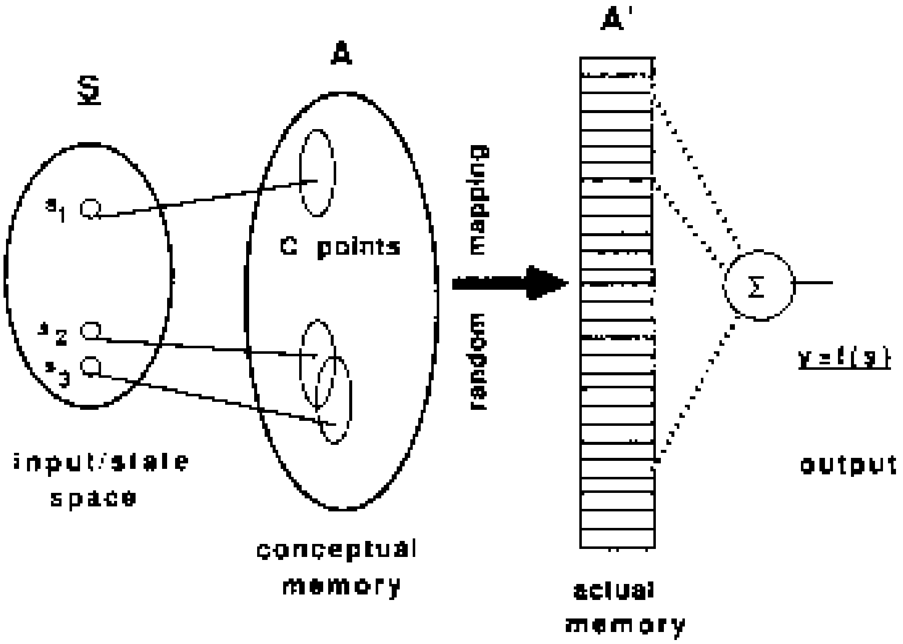


Figure 10-24 An overview diagram of CMAC.

The CMAC algorithm maps any input it receives into a set of C points in a large "conceptual" memory (A in Figure 10-24) in such a way that two inputs that are adjacent in input space will have their C points overlap in the A memory, with more overlap for closer inputs. If two inputs are far apart in the input space there will be no overlap in their C-element sets in the A memory, and therefore no generalization would be possible.

For practical systems the input space is extremely large. For example, a system with 10 inputs, each of which can take on 100 different values, would have 10010 = 1020 points in its input space, requiring a correspondingly large number of locations in the memory A. Since most learning problems do not involve all of the input space, the memory requirement is reduced by mapping the A memory onto a much smaller physical memory A'.

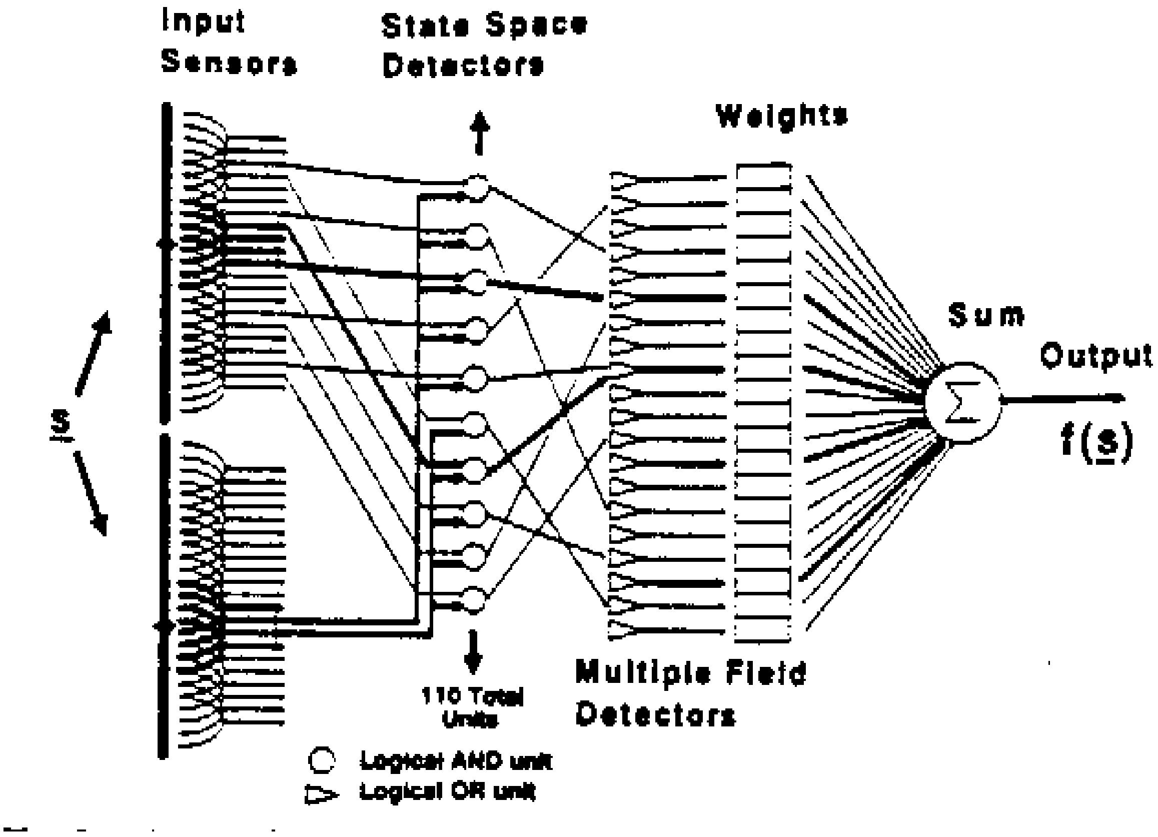


Figure 10-25 A simple example of a CMAC neural network with two inputs and one output.
The generalization parameter C has the value 4.
(Only a partial set of the state space detectors is shown).

As a result, any input presented to CMAC will generate C real memory locations, the contents of which will be added in order to obtain an output. Notice that the nonlinearity, expected from all neural networks, is in the associative input mapping, not in the sigmoid/threshold function normally at the output of each neuron.

Figure 10-25 demonstrates a network diagram of a two-input CMAC as implemented in the laboratory of the U. of New Hampshire. Each variable in the input state vector s is fed to a series of input sensors with overlapping receptive fields. Each input sensor produces a binary output which is ON if the input falls within its receptive field and is OFF otherwise. The width of the receptive field of each sensor produces input generalization, while the offset of the adjacent fields produces input quantization. Each input variable excites exactly C input sensors, where C is the ratio of generalization width to quantization width (C =4 in Fig. 10-22, C=[from 32 to 256] in typical implementations).

The binary outputs of the input sensors are combined in a series of threshold logic units (called state-space detectors) with thresholds adjusted to produce logical AND functions (the output is ON only if all inputs are ON). Each of these units receives one input from the group of sensors for each input variable, and thus its input receptive field is the interior of a hypercube in the input hyperspace (the interior of a square in the two-dimensional input space of Figure 10-25). The state space detectors of Figure 10-25 correspond logically to the individual memory locations of the A memory in Figure 10-24.

If the input sensors were fully interconnected, there wouId be a very large number of state-space detectors, and a large subset of these detectors would be excited for each possible input. The input sensors are interconnected in a sparse and regular fashion, however, in such a way that each input vector excites exactly C state-space detectors. The total collection of state-space detectors is divided into C subsets. The receptive fields of the units in each of the subsets are organized so as to span the input space without overlap. Each input vector excites one state space detector from each subset, for a total of C excited detectors for any input. There are many ways to organize the receptive fields of the individual subsets which produce similar results. In our implementation, each of the subsets of state-space detectors is identical in organization, but each subset is offset relative to the others along hyperdiagonal s in the input hyperspace (adjacent subsets are off set by the quantization level of each input variable).

Grouping in the Compressed Tessellated State Space. The organization of the receptive fields of the state-space detectors guarantees that a fixed number C of detectors is excited by any input. However, the total number of state space detectors can still be large for many practical problems. On the other hand, it is unlikely that the entire input state-space of a large system would be visited in solving a specific problem (most of the possible input vectors would never be experienced). Thus, full state space is not necessary to store unique information for each receptive field. Following this logic, the outputs of the state-space detectors are connected randomly to a smaller set of threshold logic units (called multiple field detectors in Figure 10-22) with thresholds adjusted such that the output will be ON if any input is ON (a logical OR function). The receptive field of each of these units is thus the union of the fields of many of the state space detectors. Since exactly C state-space detectors are excited by any input, at most C multiple field detectors will be excited by any input. The converging connections between the large set of state-space detectors and the smaller set of multiple field detectors are referred to as "collisions." In practice, the converging connections are implemented by assigning a virtual address to each of the state-space detectors, and passing the addresses of the active state-space detectors through a random hashing function.

Finally, the output of each multiple field detector is connected, through an adjustable weight, to an output summing unit. The output for a given input is thus the sum of the weights selected by the excited multiple field detectors. The set of all weights in Figure 10-25 corresponds to the A' memory of Figure 10-24. Note that while the number of input sensors and state-space detectors is determined by the number of inputs, their dynamic ranges, the level of quantization, and the degree of generalization, the number of multiple field detectors and adjustable weights is an independent design parameter. The necessary size of the weight memory is related to the size of the subset of the input space that is likely to be visited in solving a particular problem, rather than to the total size of the input space. Thus, simple problems in a many dimensional input space require only small weight memories, even though the size of the input space may be huge.

Ideally, the associative mapping within the CMAC network assures that nearbv points in the input space generalize while distant points do not. The effect of the converging connections between the state-space detectors and the multiple field detectors, however, is to create randomly distributed low-magnitude generalization with distant points in the space. Figure 10-26 illustrates this effect for a simple two-input CMAC. The size of the input space in this example is 128 points along the horizontal axis and 96 points along the vertical axis, for a total of 12288 points in the input space. The figure shows the degree to which a central point in the space generalizes with all other points in the space, for a CMAC with 1000 adjustable weights and c = 16. Note that even when using this small memory (relative to the size of the input space), the characteristic of local generalization in the space is largely preserved. Note also that the generalization region is not symmetrical about the horizontal or vertical axes. One can see that generalization of a single point within a 128 X 96 input space, using a CMAC with C = 16 and 1000 memory locations. Greater generalization is indicated by darker shading.

Network training is typically based on observed training data pairs of the input and the desired output (supervised learning), using the least mean square (LMS) training rule [28].

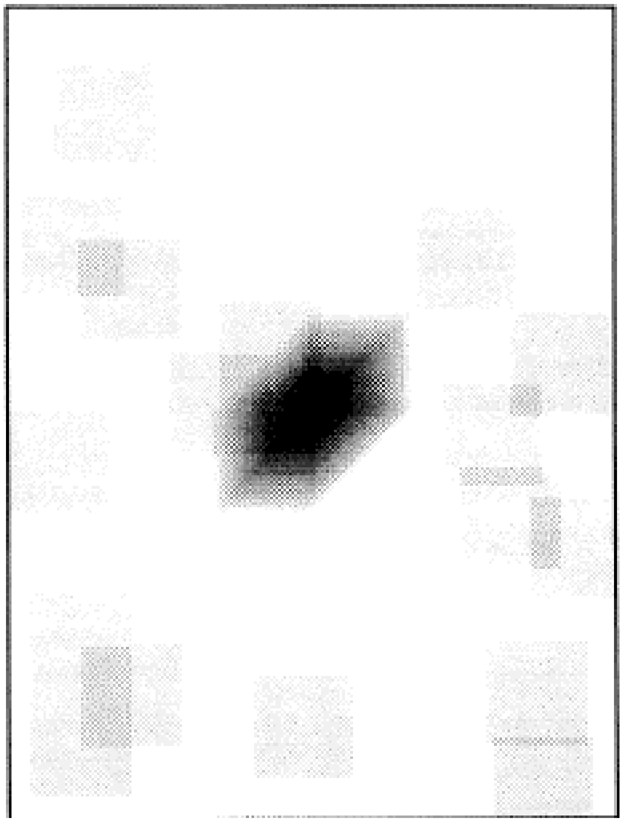


Figure 10-26 Generalization of a single point within 128x96 input space

Properties of CMAC. This sub-section summarizes the main properties of CMAC.

1) CMAC accepts real inputs and gives real outputs. The input components are quantized, but the number of levels can be as large as desired so that high degrees of accuracy is achievable.

2) CMAC has a built-in local generalization, meaning that input vectors that are "close" in the input (state) space will give outputs that are close, even if the input has not be trained on, as long as there has been training in that region of the state-space. The measure of "closeness" is Hamming distance (the sum, over all components, of the absolute value of the differences of each component). Locally generalizing networks have less learning interference than globally generalizing networks such as a multilayer perceptron.

3) CMAC has the property that large networks can be used and trained in practical time, even with the software version of the system. This is because there is a small number of calculations per output even though there is a large number of weights. In CMAC realizations at UNH, in avarage tens or hundreds of thousands of weights and 10 to 128 additions per output were employed. For both hardware and software this is a small amount of computation in comparison to the one required in an equivalent multilayer perceptron. As an example, in a pattern recognition problem CMAC required about 50 iterations while the multilayer network required about 12,000 iterations. Similar results were obtained in [27].

4) CMAC uses the LMS adaptation rule of Widrow-Hoff [4]. This least squares algorithm is equivalent to a gradient search of a surface which is quadratic and therefore has a unique minimum [28].

5) CMAC can learn a wide variety of functions. It is easy to show, that a one-input CMAC can learn any discrete one-dimensional single-valued function, given a few mild conditions on the parameters of the CMAC.

6) CMAC obeys superposition in the output space, which means that a multidimensional discrete Fourier series

can be used to show that a broad class of functions is learnable.

Problems which Require Functioning of CMAC. CMAC is an adaptive system which allows for a broad class of control functions to be computed by referring to a Look-up-Table rather than by mathematical solution of simultaneous equations (for a case with many degree's of freedom). CMAC can combine the variables into an input vector which is used to address a memory where the appropriate output variables are stored. Each address consists of a set of physical memory locations, the arithmetic sum of whose contents is the value of the stored variable. The memory addressing algorithm takes advantage of the deterministic nature of the control function: it allows for finding a unique correspondence of this type. CMAC does it in a way that promises to make it possible to store the necessary data in a physical memory of practical size.

In order to carry out any control problem, it is necessary to drive the system through a sequence of states as a function of time. The control input to each sub-system actuator is, in general, a function not only of time, but of other state variables as well. They depend on higher level input variables that identify the particular task to be performed as well as on many parameters and variables originating in the external environment. In order to deal with a problem of this complexity without either sidestepping the computational difficulties, as in direct human control systems [32, 33] one should either ignore most of the relevant variables, as in point-to-point industrial robot control [34], or it is necessary to partition the control problem into manageable sub-problems [35], or both (see Chapter 7).

For controlling intelligent systems, the computations required to coordinate individual subsystems so as to produce a particular motion trajectories of sub-systems are usually solved by computations based on concrete relationships between architectural sub-systems of IS (for example, see [36, 37]). In the resolved motion rate control system, end-effector motion is expressed as a function of all the individual joint motions. The computations performed for this are typically based on incomplete and erroneous mathematical approximations (the difficulties in representing real factors are described in [35, 38]. As many factors are introduced, the formalisms of systems of this type become less and less tractable. It is simply not possible to deal with many degrees, of flexing and twisting or a very broad range of force, touch, and acceleration inputs by systems of simultaneous equations which can be solved by computer programs of practical speed and size.

When one examines the type of manipulation tasks routinely performed by biological organisms such as squirrels jumping from tree to tree, birds flying through the woods, and human playing tennis or football, one is left with the distinct impression that the solution of mathematical equations is a totally inadequate method for producing truly sophisticated motor behavior. it seems clear that the present mathematical formalisms for manipulator control are in deep trouble when addressing the type of mechanical control problems which are obviously trivial for the brain of the tiniest bird or rodent.

Cerebellum as the Inspiration for the Neural Network Learner. This is not to suggest that the proper course for research in the IS control should be to attempt to model the structural properties of the biological brain. Early attempts along these lines were notoriously unsuccessful in producing any significant results and the subsequent disillusionment has strongly prejudiced the intellectual community against seeking any guidance from the numerous existence theorems provided by nature. We believe, however, that it may be possible to duplicate the *functional* properties of the brains without modeling the structural characteristics of the neuronal substrate.

One part of the brain that seems to be intimately involved in generating motion trajectories is the cerebellum. Recent anatomical and neuro-physiological data has led to a detailed theory concerning the functional operations carried out by the cerebellum [39, 40]. Input to the cerebellum arrives in the form of sensory and proprioceptive feedback from the muscles, joints, and skin together with commands from higher level motor centers concerning what motion trajectories are to be performed. According to the theory, this input constitutes an address, the contents of which are the appropriate muscle actuator signals required to carry out the desired movement. At each point in time the input addresses an output which drives the muscle control circuits. The resulting motion produces a new input and the process is repeated. The result is a trajectory of the limb through space. At each point on the trajectory the state of the limb is sent to the cerebellum as input, and the cerebellar memory responds with actuator signals which drive the limb to the next point on the trajectory.

A neuro-physiological theory of how the cerebellum accomplishes these tasks has been published in [41, 42]. This sub-section describes the mathematical concepts of how the cerebellum structures input data, how it computes the addresses of control signals, how the memory is organized, and how the output control signals are generated. Certain features of the neurophysiological and anatomical structure of the cerebellum has led to the theory [40]

many respects to a Perceptron [43]. The Perceptron is a member of a whole family of trainable Pattern-classifying machines, or machines which distinguish between patterns on the basis of linear discriminate functions [44].

The CMAC Mapping Algorithm. The CMAC algorithm functions by breaking the

$$S \rightarrow A$$

mapping into two sequential mappings

$$S \rightarrow M$$

and then

$$M \rightarrow A$$

Each R-ary variable s_i in the input vector $S_i = (s_1, s_2 ' \ldots s_N)$ is first converted into a binary variable m_i according to the following rule:

The Rule of Conversion:

1. Each digit of the binary variable m_i must have a value of "1" over one and only one interval within the range of S_i and must be "0" elsewhere.

2. There are always IA*l equal to “1” in the binary variable m_i for every value of the variable S_i.

The names of the subscripts of the binary digits in m* are then tabulated against the values of the variables s.

Multidimensional mappings. The complete mapping S →M consists of N individual mappings $S_i \rightarrow m^*_i$ for all the variables in the input vector $S_i = (s_1, s_2\,' \ldots s_N)$.

$S_1 \rightarrow m^*_1$

$S_2 \rightarrow m^*_2$
$S \rightarrow M$ …
…
$S_N \rightarrow m^*_N$

Mapping Into a Memory of Practical Size. Thus far, we have described a means of performing the mapping f:S → A in a manner which is well suited to producing generalization where generalization is desired, and dichotomization where that is desired. We have not, however, explained how this transformation can be accomplished with a reasonable number of association cells. The concatenation of -subscript names m*i produces a potentially enormous number of association cell names. If each variable in has R distinguishable values, then there are *R* distinguishable points in input-space. After the mappings, concatenation to obtain A* yields a potential number of association cell names on the same order of magnitude as *R.* For any practical manipulator control problem, the number of input variables N is likely to exceed 10, and the number of distinguishable values R of each variable will probably be 30 or more. The number 3010 is clearly an impossibly large number of association cells for any practical control device. If, however, it is not required for input vectors outside of the same neighborhood to have zero overlap, but merely a small probability of significant overlap, then it is no longer necessary to have *RN* association cells. Assume that an additional mapping A --+ A, is performed such that the *RN* association cells in the very large set A are mapped onto a much smaller, physically realizable set A, One way in which this can be done is by hash-coding [141.

Hash-coding is a commonly used computer technique for reducing the amount of memory required to store sparse matrices and other data sets where a relatively small amount of data is scattered over a large number of memory locations. Hash-coding operates by taking the address of where a piece of datum is to be stored in the larger memory and using it as an argument in a routine which computes an address in the smaller memory. For example, any address in the larger memory might be used as an argument in a pseudo-random number generator whose output is restricted to the range of integers represented by the addresses in the small memory. The result is a many-to-few mapping of locations in the larger memory onto locations in the smaller. Any, association cell name (address) in A can be used as the argument in a hash-coding routine to find its counterpart in A'. The number -of association cells in A, can be chosen arbitrarily equal to the size of the physically available memory. In practice A' may be orders-of-magnitude smaller than A. Thus, the A → A' mapping is a many-into-few mapping.

The many-into-few property of the hash-coding procedure leads to "collision" problems when the mapping routine computes the same address in the smaller memory for two different pieces of data from the larger memory. Collisions can be minimized if the mapping routine is pseudo-random in nature so that the computed addresses are as widely scattered as possible. Nevertheless, collisions are eventually bound to occur, and a great deal of hash-coding theory is dedicated to the optimization of schemes to deal with. CMAC, however, can simply ignore the problem of hashing collisions because the effect is essentially identical to the already existing problem of cross-talk, or learning interference, which is handled by iterative data storage.

Inputs From Higher Levels. Commands from higher centers are treated by CMAC in exactly the same way as input variables from any other source. The higher level command signals appear as one or more variables in the input vector S. Reference variable affecting the selection of A*,. The result is that input signals from higher levels, like all other input variables, affect the output and thus can be used to control the transfer function. If, for example, a higher level command signal changes its value then the sample set will change. If the change is large enough then the concatenation process will converge to a different result.

Thus, by changing the signal the higher level control signal can effectively change the CMAC transfer function. This control can either be discrete or continuously variable (i.e., it can vary smoothly over its entire range). An example of the types of discrete commands which can be conveyed to the CMAC by higher level input variables are "reach," "pull back," "twist," "scan along a particular surface," etc. (see [11]). An example of the types of continuously variable commands which might be conveyed to the CMAC are velocity vectors describing the motion components desired of the manipulator end-effector. Three higher level input variables might be representing the commanded velocity components of a manipulator end-effector in a coordinate system defined by some work space.

Conclusions. In the foregoing we have explained the properties of a neural network called CMAC and described its mechanism. This form of neural network has advantages and disadvantages in comparison to other forms of neural networks. CMAC has the advantages that it is fast in software and can be realized in high speed hardware, thereby leading to the practical use of more weights and larger systems in solving problems. Furthermore, CMAC is able to learn a large variety of nonlinear functions, reducing the need to use slow backpropagation based networks. CMAC provides for a local generalization which may have advantages over global generalization: There is little or no learning interference due to recent learning in remote parts of the input space, and local generalization usually requires a smaller number of additions, therefore giving fast computation speeds. Furthermore, CMAC/LMS training takes fewer iterations than back propagation. The speed and network size advantage of CMAC means that real systems and real problems can be undertaken with results beyond the quality of other standard methods.

On the other hand, CMAC has some disadvantages: The generalization is not global, so we will not find mysterious properties "emerging" from the training, if that is an advantage; collisions due to the hash coding necessary to reduce the memory size to something realizable can cause a noise or interference if care is not taken in design to prevent that happening; and some design care must be exercised in order to be assured that a low error solution will be learned in a specific application.

One can conclude that the CMAC neural network is available alternative to the multilayer backpropagated network for learning situations with analog inputs and outputs, especially where speed of convergence and speed of computation are important, and where large numbers of weights are needed. CMAC has proven able to learn unknown nonlinear functions quickly and generalize on inputs it has never seen.

CMAC computes control functions by referring to a Look-up-Table rather than by solution of analytic equations or by other conventional techniques. Functional values are stored in a distributed fashion such that the value of the function at any point in input-space is derived by grouping together the contents over a number of memory locations.

The unique feature of CMAC is the mapping algorithm that converts distance between input vectors into the degree of overlap between sets of addresses where the functional values are stored. CMAC is thus a memory management generalizing technique which causes similar inputs to tend to group into new units so as to produce similar outputs; yet dissimilar inputs result in outputs which are independent.

References:

[1] J. S. Albus, "A theory of cerebellar functions," Mathematical Biosciences, Vol. 10, pp. 25-61,1971.

[2] J. S. Albus, "Theoretical and experimental aspects of a cerebellar model," Ph.D. dissertation, Univ. of Maryland, 1972.

[3] Rosenblatt, *F., Principles of Neurodynamic8: Perceptrons and the Theory of Brain Mechanisms,* Spartan Books, Washington. *D.C., 1961*

[4] B.Widrow, M. E. Hoff, "Adaptive switching circuits," *Proc. IRE Western Electronic Show and Conv.*, Vol. 4, pp. 96-104, *1960.*

[5] B. Widrow, "Generalization and information storage in networks of adaline neurons”, *Self-Organizing Systems*, M. C. Yovits, Ed. Wash., DC: Spartan, 1962, pp. 435-%1.

[6] J.J. Hopfield, "Neuronswith graded response have collective computational properties like those of two-state neurons," *Proc. Natl. Acad. Sci. USA,* Vol. 81, pp. 3088-3092, May 1984,

[7] J.J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nad. Acad Sci. USA,* Vol. 79, pp. 2554-2558, Apr. 1982.

[8] J. S. Albus, "Data storage in the cerebellar model articulation controller," 1. *ASME Transactions on ynamic* systems, *Measurement and Control, pp. 228-233,* Sept. 1975.

[9] J. S. Albus, "A new approach to manipulator control: *the* cerebellar model articulation controller (CMAC)," pp. 220-227, Sept. 1975.

[10] W. T. Miller, "A nonlinear learning controller for robotic manipulators," *Proc. SPIE, Intelligent Robots and Computer* Vision, Vol. *726, pp. 416-423,* Oct. 1986.

[11] W. T. Miller, "A learning controller for nonrepetitive robotic operations," *Proc. Workshop on Space Telerobotics,* publication 87-13, Vol. 11, Pasadena, CA, pp. 273-281, Jan. 10-22,1987.

[12] W. T. Miller, "Sensor based control of robotic manipulators using a general learning algorithm," ***IEEE** Trans. Robotics Automat.*, Vol. RA-3, pp. 157-165, Apr. 1987.

[13] W. T. Miller, F. H. Glanz, and L G. Kraft, "Application of a general learning algorithm to the control of robotic manipu I ato rs, " *Internat. J. Robotics Research,* Vol. 6, p p. 84-98**,** Summer 1987.

[14] W.T. Millerand R. P. Hewes, "Real timeexperiments in neural network based learning during high speed nonrepetitive robotic operations," *Proc. 3rd IEEE Int. Symp. on Intelligent* Control, pp. 513-518, Aug. 24-26,1988.

[15] W. T. Miller, "Real time learned sensor processing and motor control for a robot with vision," *1st Annual Conf. of the Intl. Neural Network Society, p. 347, Sept. 1988.*

[16] W. T. Miller, "Real time application of neural networks for sensorbased control of robots with *vision," IEEE SMC,* Vol. 19, pp. 825-831, July-Aug. 1989.

[17] F. H. Glanz and W. T. Miller, "Shape recognition using a CMAC based learning *system," Proc. SPIE ConL on Robotics and Intelligent Systems, vol. 848, pp. 294-298, Nov. 1987.*

[18] F. H. Glanz and W. T. Miller, "Deconvolution using a CMAC neural network," Proc. *1st Annual Conf. of the Intl. Neural Network Society, p. 440,* Sept. 1988.

[19] F. H. Glanz and W. T. Miller, "Deconvolution and nonlinear inverse filtering using a neural network,"' Intl. Conf. on *Acoustics and Signal Processing,* Vol. *4, pp. 2349-2352,* May 23-29,1989.

[20] L G. Kraft and D. P. Campagna, "A comparison of CMAC neural network and traditional adaptive control systems," *Proc. 1989 American Control Conf.,* Vol. 1, pp. 884-889, June 21-23,1989.

[21] D. Herold, W. T. Miller, L G. Kraft, and F. H. Glanz, "Pattern recognition using a CMAC based learning system," Proc SPIE, *Automated Inspection and High Speed Vision Architectures 11,* Vol. 1004, pp. 84-90, Nov. 10-11, 1989.

[22] W. T. Miller, R. P. Hewes, F. H. Glanz, and L. G. Kraft, "Realtime dynamic control of an industrial manipulator using a neurai-network-based learning controller," *IEEE Trans. Robotics Automat.,* Vol. 6, pp. 1-9, Feb. 1990.

[23] E. Ersu and H. Tolle, "Hierarchical learning control-An approach with neuron-like associative memories," Proc. *IEEE Conf. on Neural Information Processing Systems, Nov. 1988.*

[24] E. Ersu and H. Tolle, "A new concept for learning control inspired by brain theory," *Proc. FAC 9th World Congress, July 2-6, 1984.*

[25] E. Ersu and J-Militzer, "Real-time implementation of an associative memory-based learning control scheme for non-linear multivariable processes," *Proc. 1st Measurements and Control Symp. on Applications of Multivariable Systems Techniques, pp. 109-119, 1984.*

[26] E. Ersu and X. Mao, "Control of pH using a self-organizing control concept with associative memories," Proc. Intl. *IASTED Conf. on Applied Control and Identification, 1983.*

[27] J. Moody, "Fast learning in multi-resolution hierarchies," in *Advances in Neural Information Processing, D.* Touretzky, Ed. Morgan Kauimann, *1989.*

[28] B. Widrow and S. D. Stearns, *Adaptive Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, *1985.*

[29] W. T. Miller, B. A. Box, and E. C. Whitney, "Design and implementation ot a high speed CMAC neural network using programmable CMOS logic cell arrays," Univ. *of New* Hampshire, Rept. no. ECE.IS.90.01, Feb. *6, 1990.*

[30] K. Hardwicke, "On the applicability of piecewise affine models in nonlinear adaptive equalization," MS thesis, University of Texas at Austin, *1987*

[31] W. T. Miller, III, F. H. Glanz, L. G. Craft, “CMAC: An Associative Neural Network Alternative to Backpropagation”, Proc. IEE, Vol. 78, No. 10, Oct. 1990, pp. 1561-1567

[32] Johnsen, E. G., Corliss, W. R., *Teleoperators and Human Augmentation,* NASA SP-5047, 1967

[33] Corliss, W. R., and Johnsen, E. G., *Teleoperator Controls,* NASA *SP-5070, 1968*

[34] Ashley, J. R., and Pugh, A., "Logical Design of Control Systems for Sequential Mechanisms," *International Journal of Production Research (I.P.E.), Vol. VI,* November 4, 1968

[35] Paul, R., "Modeling, Trajectory Calculation and Servoing of A Comi uter Controlled Arm." PhD thesis. Stanford Universitv, *1972*

[36] Whitney, D. E., "Resolved Motion Rate of Manl*vulat rs and Human Prostheses," *IEEE Transactions on Man-Machine System& Vol.* MMS-10, No. 2, June 1969, pp. 47-53

[37] First Annual Report for the Development of MultiModed Remote Manipulator Systems, Charles Stark Draper Laboratory (Division of Massachusetts Institute of Technology), Report C-3790

[38] Kahn, M. E., and Roth, B., "The Near Minimum-Time Control of Open Loop Articulated Kinematic Chains," JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL, TRANS. ASME, Series G, Vol. XCIII, No. 3, Sept. 1971. D - 164-172

[39] Grossman, S. P., "The Motor System an'Afechanics of Basic Sensory-Motor Integration," Chapter 4 of *Textbook of Physiological Psychology,* Wiley, New York, 1967

[40] Albus, J. S., "A Theory of Cerebellar Function," *Mathematical Bio8ciences, Vol. X,* 1971

[41] Albus, J. S., "A Robo;%ond'itioned Reflex System Modeled After the Cerebellum." *Proceedinas Fall Joint Computer Conference, Vol. XLI,* 1972, pp. 1095-1104

[42] Albus, J. S., "Theoretical and Experimental Aspects of a Cerebellar Model," PhD thesis, University of Maryland, Dec. 1972

[43] Werbos, P. J. “Generalization of backpropagation with application to a recurrent gas market model”. *Neural Networks*, 1, 1988, pp. 339-356