CONF-900729

DISCLAIMER

EXPERIMENTATION AND CONCEPT FORMATION BY AN AUTONOMOUS MOBILE ROBOT*

CONF-900729--1

DE90 003063

by

P. F. Spelt, G. deSaussure,
G. Oliver, and M. Silliman

Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6364

Submitted to: Japan-USA Symposium on
Flexible Automation

July 9 - 11, 1990
Kyoto, Japan

* Research sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, of the U. S. Department of Energy, under contract No. DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

EXPERIMENTATION AND CONCEPT FORMATION BY AN AUTONOMOUS MOBILE ROBOT*

by

P. F. Spelt, G. deSaussure,
G. Oliver, and M. Silliman

Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6364

The Center for Engineering Systems Advanced Research (CESAR) conducts basic research in the area of intelligent machines. In this paper, we describe our approach to a class of machine learning which involves autonomous concept formation using feedback from trial-and-error experimentation with the environment. Our formulation was experimentally validated on an autonomous mobile robot, which learned the task of control panel monitoring and manipulation for effective process control. Conclusions are drawn concerning the applicability of the system to a more general class of learning problems, and implications for the use of autonomous mobile robots in hostile and unknown environments are discussed.

I. Introduction

The Center for Engineering Systems Advanced Research (CESAR) has recently undertaken several research activities in the field of machine learning. This paper describes work in autonomous learning using HERMIES-IIB, the third in our series of robotic experimental testbeds. The integrated system in HERMIES-IIB (Hostile Environment Robotic Machine Intelligence Experiment Series IIB) is designed to ultimately perform in environments which humans cannot readily enter. A detailed description of this machine and its navigation capabilities has appeared in IEEE Expert.[1] Briefly, the computing power consists of two components -- a VME subsystem for vision input and for the I/O devices, and an IBM 7532 (an industrialized PC-AT) for the "brain". Four AT expansion slots house boards which provide an onboard 16-node NCUBE hypercube parallel computer. The hypercube machine is used for both vision processing and for running the Expert Systems used for navigation and learning.

Research using the HERMIES series of robots emphasizes computational autonomy, with all processing done using the onboard computer system described above. To date, research has focused on navigation in a dynamic environment, including the ability of the robot to deal with unexpected moving obstacles using any of several strategies. These strategies include replanning the goal path, moving small obstacles out of the way, or waiting until moving obstacles have cleared the robot's path[1]. The robot's navigational goal is to position itself in front of a process control panel, enabling it to read meters and manipulate buttons and levers. Here we describe a system which learns the control panel's system dynamics and remembers the most efficient series of responses to "shut down" a control process, for future encounters with similar (but not necessarily identical) situations. The system also infers a classification scheme for panel categories, enabling it to hypothesize about correct response sequences for new panel configurations. The system includes a hypothesis-generating scheme which permits the robot to make a "best guess" about a response sequence for panels it is unable to classify.

II. Machine Learning Background

Carbonell[2] recently argued that machine learning is central to all areas of the field of artificial intelligence, and is defensibly a prerequisite to any form of intelligence. In this context, he identified four major paradigms, each with its own subdivisions. These are inductive learning, analytic learning (explanation-based learning), genetic algorithms, and artificial neural networks. The first two are qualitative, symbol oriented types; the latter two are quantitative, mathematical types. Michalski[3] identified five modes by which machines can transform information in the learning situation: rote learning, learning by instruction, learning by deduction, learning by analogy, and learning by induction, listed in increasing order of complexity of inferencing on the part of the learner (3, p.14). Learning by induction was further divided into learning by observation and discovery, and learning from examples, with the examples provided either by a knowledgeable teacher or by the environment. When the environment does the teaching, the learner performs experiments from which it receives feedback on the correctness of the performance, with each experiment resulting in an example from which the learner can generalize. For this instance-to-class generalization, the learner uses these experiment-generated examples, inferring from them a general classification scheme which describes the classes.

Spelt[4] has presented a broader conceptualization of both biological and machine learning paradigms, arranged in a hierarchical form which emphasizes the fact that so-called "higher" cognitive functions (e.g., concept formation and problem solving) are based on more elementary forms of learning (e.g., operant conditioning and discrimination learning). Such a scheme holds that learning paradigms higher in the hierarchy build on the knowledge and skills acquired through types of learning which fall lower in the hierarchy.

Most of the recent research published on machine learning has focused on the higher cognitive functions (problem solving, concept formation, rule learning, etc.; see 3, 5, and the journal Machine Learning), with almost none appearing in the robotics literature. This interest in higher cognitive abilities understandably stems from the perception that these are uniquely human attributes directly associated with "Intelligent Behavior". However, such exclusively cognitive tasks require no motor behavior capabilities such as our robot displays -- one needs only a stationary "electronic brain" to do the information processing. McMillan[6], whose work provides an exception to this trend, discussed other learning paradigms which might serve as helpful models for learning in intelligent machines. Some of these models are especially useful for work with an autonomous mobile robot, which can move around and manipulate the environment. McMillan's work simulated a low-level learning paradigm (Classical or Pavlovian Conditioning) which was used to adjust an operating system's presentation of a command menu. The new sequence selects an anticipated next command, based on the user's past use of those commands. Similarly, Laird, Rosenbloom and Newell[7] have developed a system called SOAR, which exhibits automated learning in a wide variety of tasks including motor performance, and which takes biological models as a source for some of its concepts.

Operant Conditioning (also known as Trial-and-Error Learning), involves the manipulation of objects in the environment with such behavior followed by some type of feedback (reinforcement or punishment) concerning the suitability of the responses in that setting. Examples include the use of a wide variety of ON/OFF switches on appliances and machinery by humans, and the cracking open of many shelled sources of food such as nuts and oysters by animals). In robotics work, an approach by Mitchell[8] uses feedback from the environment to guide learning by a manipulator to skillfully move objects around in an incompletely known environment. Mitchell's system is an application of explanation-based learning to robotics, in which the system applies domain knowledge to explain response failures.

Because much of so-called higher human learning is based on these simpler forms of conditioning at various points in the human's learning history, it seems useful to explore such learning for autonomous robots, as a foundation for more cognitive kinds of robot activities. Therefore, our system was designed to perform learning-by-induction using instance-to-class generalization from examples the robot has provided for itself by experimenting with the CESAR process control panel. The system presented here forms concepts for classifying control panel examples on the basis of knowledge acquired through trial-and-error learning provided either by a computer simulation or by direct experience of the robot with the control panel.

III. The Learning Expert System

The problem space the robot learned was comprised of 81 panel configurations, each defined by a combination of initial settings of 2 meters (high, middle or low) and 2 levers (right, middle or left). These 81 panel states are grouped into a set of categories initially defined by the response sequence which terminates the "Danger" light on the panel. Each category has a constant set of defining attributes based on a subset of initial meter and lever settings. Different categorization schemes can be created by using different combinations of attribute settings, each coupled with its unique response string[9].

The Learning Expert System was implemented as a rule-base in the expert system shell CLIPS[10]. The rule-base consists of three major components: a Hypothesis-Generating unit which permits the robot to hypothesize about possible correct solutions for previously unseen problems; a Response-Sequence Learning unit which learns sequences of responses on problems for which the robot cannot generate a correct hypothesis; and an inferencing (Category Formation) component to generalize from the examples, enabling the robot to infer categories of problems. The rule-base underwent several major revisions during development. Data from three stages are presented in the evaluation section. However, only the final rule-base is described here. Important differences between this version and earlier ones are mentioned in the results section.

A. Hypothesis Generation

The first step in solving a new panel problem is Hypothesis Generation, shown in Figure 1. If no match is possible, the system defaults to the Response Selection process described in the next section. If a match is possible, the first determination is whether it is an exact match or one which places the new panel problem into an existing category. In either case, the response sequence associated with the panel or category is proposed as the solution to the new problem.

Insert Fig. 1 about here

In cases where there is a partial match, but no categorization is possible, the system calculates a "Best Guess". This process assigns weighted values to attributes of the new problem which match, to varying degrees, the attribute values of each memory vector stored. The response sequence of the vector with the highest weight is proposed as the solution to the current problem. Should that sequence fail at some point, the system selects the next most highly-weighted sequence, picking up at the point at which the previous sequence failed. Thus, the "Best Guess" process should provide greater response-learning efficiency in situations where the robot has some previous experience, but not enough to create accurate category descriptions. This is most likely to occur early in training, before many categories have been defined.

B. Response Selection and Learning

Response selection occurs whenever the robot confronts a new panel problem, with the particular selection strategy depending on the availability and type of prior experience (see Figure 2). If no prior experience exists, responses are selected from the arbitrarily ordered list, as described below. Otherwise, the system should be able to generate a hypothesis about what the correct solution might be, as described in part A of this section.

Insert Fig. 2 about here

The Response-Sequence Learning unit[10] operates when the naive robot has no past experience to guide its behavior at the panel. It consists of a subset of the rule base which discovers, by trial-and-error learning, the appropriate sequence of control panel button and lever actions to solve problems represented by the values of various attributes (the initial settings of the 2 meters and the 2 levers) of the panel. As indicated at the top of Figure 2, this sequence learning process involves a breadth-first search through an arbitrarily-ordered list of responses to discover which one is appropriate at a particular point in the problem-solving process. In the case where there is some past experience but no reasonable hypothesis can be created (as when there is only one example for various categories), the robot can choose from a table of weighted responses. This weighting, compiled across all classification schemes the robot has experienced, is a simple tabulation of the frequency with which a particular response has occurred at a given point in previous sequences.

Whichever response selection method the robot uses, it tries the selected response on the control panel and examines the consequences. A correct response receives immediate feedback, and the system then adds that component to the sequence being built. An incorrect response receives no feedback, and the robot then selects the next response to try. Once the entire correct sequence has been determined (indicated by extinguishing the "Danger" light), the system then associates that response sequence with the initial set of panel attributes in which the sequence was learned.

The result of the panel manipulation process just described is a set of memory vectors, one per panel problem solved, each containing a coding of the initial panel state associated with a 4-unit response sequence which turned off the "Danger" light for that panel configuration. Thus, this process learns the basic category examples, consisting of these initial environmental states and associated response chains, which form the basis for inferring problem categories by the Inferencing, or Concept Formation part of the system.

C. Autonomous Category (Concept) Formation

The third component of the rule base is the Category Inferencing unit, detailed in Figure 3. This is a similarity-based (3, p 16) generalization system which explores examples and counter-examples of a category (inter-example relationships) to create concept descriptions. It searches for attribute values shared by examples in the same class and ignores those that are different. It also identifies those attribute values which are different among the different categories, permitting discrimination among those categories.

Insert Fig. 3 about here

As illustrated in Figure 3, there are four steps to defining categories. First, the system assumes that all panels which operate the same way belong to the same category, and therefore, all memory vectors are grouped according to the response sequence which turned off the Danger light. Second, the attributes of the initial states for one pair of memory vectors from a group are parsed to identify both those attributes which have a common value and those which do not. Dissimilar attributes within a group can be ignored, unless subsequent experience indicates to the robot that they belong to a logically complementary category, such as a meter being low or middle but NOT high. Once common attribute values have been identified in a pair of category examples, all other examples from that category are parsed for consistency with the category descriptors. At this point, adjustments to the descriptors are made to achieve an accurate category description as, for example, in the case of the logical complement descriptors mentioned for the NOT-high meter. This condition requires extra processing on an additional example in order to establish the precise conditions under which examples will fit the category or be excluded.

The final step in establishing category descriptions is to check among the various groups to make sure the descriptors can differentiate among the categories as defined by the response sequences. The process involves checking the defining attributes to be sure their values permit discriminating among the categories. This is done sequentially, the first attempt being to discriminate on the basis of one attribute. If this does not succeed, then pairs of attributes are considered, and finally triads. Once all examples in the robot's current working memory have been processed, the system is ready to attempt a new panel problem.

IV. Evaluation of the Expert System

Because the proper logical operation of the rule base is independent of the robot's actual interaction with the panel, we were able to use a two-computer system which simulates the interaction between the robot and the panel[11]. This system consists of PC/XT-type machines communicating with each other through serial ports. As shown in Figure 4, the panel emulator presents the rule base a series of problems randomly selected from a batch file, and records the number of errors made on each problem. The robot simulator runs the expert system rule-base, sending commands concerning responses attempted to the panel emulator. The panel emulator also returns information concerning the current state of the panel, including feedback, to the Expert System. The data presented below were obtained using this simulator. It should be noted that the system has also been tested on the robot[9], showing the same performance as on the simulator.

Insert Fig. 4 about here

A. Experimental Procedure

Five classification schemes were devised to test the flexibility of the Expert System. In order to simplify information processing during the initial development of the expert system, the analog values of the levers and meters were mapped into the previously mentioned 3-valued logic system inside the rule base. We assumed the buttons would all be OFF when the robot first approached the panel, and the Danger light ON, signaling that the robot must do something. Our five classification schemes used different subsets of the four devices to define categories, with two of the schemes using logical complement characteristics. In a classification system with a fixed number of examples, the number of examples per category is a function of the number of defining attributes: one attribute can create three categories, one for each value of the defining attribute (low/left, middle, or high/right), creating a system with 27 examples per category. Similarly, two attributes can define 9 categories, and three can define 27. All our schemes were logically complete: all panels were categorized, and a panel could only appear in one category.

B. Results

Figure 5 presents data from three versions of the expert system rule-base, the original version (left panel), an intermediate version (middle panel) and the latest version with the BEST Guess segment (right panel). The 14 problems were presented in random order, although the data have been paired for presentation in the graphs. The problems were selected to give the rule-base 2 examples from each of the 7 categories, thus permitting complete category description after the initial 14 problems. This "friendly teacher" approach was necessary for the original rule-base to form category descriptions, a process carried out only after all 14 problems had been presented. The intermediate version (middle panel, Figure 5) processed memory vectors to obtain category descriptions any time it had two examples from one category available. The right panel in Figure 5 shows the performance of the rule-base with the Best Guess unit added.

Insert Fig. 5 about here

As can be seen in Figure 5, performance improved only slightly from the original to the intermediate version of system. However, the change from memory processing only after sufficient examples for all categories (one-shot inductive learning) to processing as much as possible after each example (incremental induction) was an important step which permitted the rule-base to make use of the Best Guess component. Addition of the Best Guess unit dramatically improved the performance of the system. As Figure 5 shows, the expert system made fewer errors with the Best Guess function than without. In fact, the only categories on which the rule-base made errors on the second category example (white bars) were those which involved the logical complement descriptors (categories 5 and 6). Moreover, because of inadvertent similarities among the response sequences, the new rule-base was more efficient on all problems except the first, in which both systems were equally naive, and the fourth, in which the new system did slightly worse on the first example. Finally, no version of the rule-base made any errors after the initial 14 problems.

Thorough testing of the Best Guess version of the rule-base involved use of a variety of random sequences for presenting the 81 panel problems, each presentation starting with a naive rule-base. Typical results from one sequence are presented in Figure 6. The graph is truncated at problem 21, as no further errors were made after problem 20. Note that errors occurred on exactly 14 problems, one pair for each of the 7 categories in this scheme, and that there were 6 correct classifications within the 20 problems required for total knowledge about the classification scheme. Also, a number of the problems had few errors, especially problems 18 and 19, as a result of the efficient performance of the Best Guess function -- even when it made a wrong classificatin, it generated some correct resposnes.

Insert Fig. 6 about here

Figure 7 presents total errors on all 81 panel problems for each of the three rule-base versions on 2 different schemes. These schemes used different numbers and different combinations of attributes to create classification schemes which had either 9 or 27 examples per category. This permitted assessment of the flexibility and robustness of the logical inferencing system in the rule-base. As can be seen in Figure 7, the rule-base made fewer errors on the classification scheme which had fewer categories (dark gray vs light gray bars), as would be expected. More importantly, the number of errors decreases with the addition of the Best Guess routine. The right bars show the performance of the system using the weighted table described earlier. This addition did not improve performance beyond the Best Guess routine. Thus, the rule-base is clearly more efficient at learning response sequences and at classifying problems as a result of the best guess function.

Insert Fig. 7 about here

V. Summary and Conclusions

We have presented a learning expert system for an autonomous mobile robot. The expert system learns from experience generated by the robot as it experiments with the environment, and generalizes from that experience to infer categories in which it can classify new problem configurations. As currently designed, the system learns a sequence of responses to alter or shut down a control process.

However, the general methodology is applicable to any situation in which a robot needs to learn a sequence of motor operations and then make inferences about a classification scheme, such as assembly or disassembly of a mechanical device.

We also described the use of a simulation system to train the robot prior to its entering a hostile or critical environment. This training, coupled with the system's ability to solve novel tasks by generalizing, eliminates the need to pre-program all possible real world situations. We showed that by presenting a set of panel configurations with proper feedback for learning, the robot can develop the capacity to handle a wide variety of unanticipated panel configurations, making a minimum number of errors. After training on a subset of the panel problems and inferring categories, the expert system makes no further errors on new panel problems from the same classification scheme.

The combination of a robot capable of learning and a simulation system to provide rapid and efficient training seems to be a viable way of creating a robot prepared to cope with a dynamic environment whose characteristics cannot be known in advance. The inability to precisely know the robot's environment precludes programming the ability to handle all problems in advance. Use of simulators to train the robot on a variety of potential emergency situations provides a rapid and efficient way of dealing with those emergencies -- one needs only to load the proper memory disk into the robot to have a specifically trained robot which can deal with that situation and continue to learn from its new experiences.

VIII. Acknowledgements

The authors wish to acknowledge the valuable help of Deanna L. Barnett in adapting CLIPS to communicate with the robot's I/O devices, and of Stephen Killough in panel design and construction. Research for this article was sponsored by the Engineering Research Program of the US Department of Energy's Office of Basic Energy Sciences, under contract No. DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc. This research was conducted in part by an appointment of the senior author to the Faculty Research Participation Program supported by the US Department of Energy and administered by Oak Ridge Associated Universities under contract No. DE-AC05-76OR00033.

References

1 Burks, B. L., _et al_., "Autonomous Navigation, Exploration,and Recognition Using the HERMIES-IIB Robot," _IEEE Expert_, Winter, 1987. pp. 18-27.

2 Carbonell, J. G. "Introduction: Paradigms for Machine Learning." _Artificial Intelligence_, 40(1-3), pp1-9, 1989.

3 Michalski, R. S., J. G. Carbonell, & T. M. Mitchell, _Machine Learning, an Artificial Intelligence Approach_, vol II, Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1986.

4 Spelt, P. F., Breakout Session: Machine Learning _via_ Experience and Human Observation. In Parker, L. E. & C. R. Weisbin, eds., 1988 Workshop on Human-Machine Symbiotic Systems Proceedings, pp 163-170, Oak Ridge National Laboratory, ORAU 89/C-140, CESAR-89/19, December 5-6, 1988.

5 Michalski, R. S., J. G. Carbonell, & T. M. Mitchell, _Machine Learning, an Artificial Intelligence Approach_, vol I, Tioga Publishing Company, Palo Alto, CA, 1983.

6 McMillan, W. W. "Classical Conditioning as an Example of Natural Models in Machine Learning," _Methodologies for Intelligent Systems_, Z. W. Ras & M. Zemankova, eds., Elsevier Publishing Company, New York, NY, 1987.

7 Laird, J. E., _et al_, "Chunking in SOAR: The Anatomy of a General Learning Mechanism," _Machine Learning_, 1986, vol. 1, no. 1, pp. 11-46.

8 Mitchell, T. M. Toward a Learning Robot. In Parker, L. E. & C. R. Weisbin, eds., 1988 Workshop on Human-Machine Symbiotic Systems Proceedings, pp 129-141, Oak Ridge National Laboratory, ORAU 89/C-140, CESAR-89/19, December 5-6, 1988.

9 Spelt, P.F., _et al_., "Learning by an Autonomous Robot at a Process Control Panel", _IEEE Expert_, Winter, 1989, pp 8-16.

10 Culbert, C. J. _CLIPS Reference Manual_, NASA Johnson Space Flight Center, distributed by COSMIC Software Information Services, University of Georgia, Computer Services Annex, Athens, GA, 30602.

11 Spelt, _et al_., "Concurrent Simulation: Development and Training of a Learning Expert System for an Autonomous Robot", in press, _Simulation_, October, 1988.

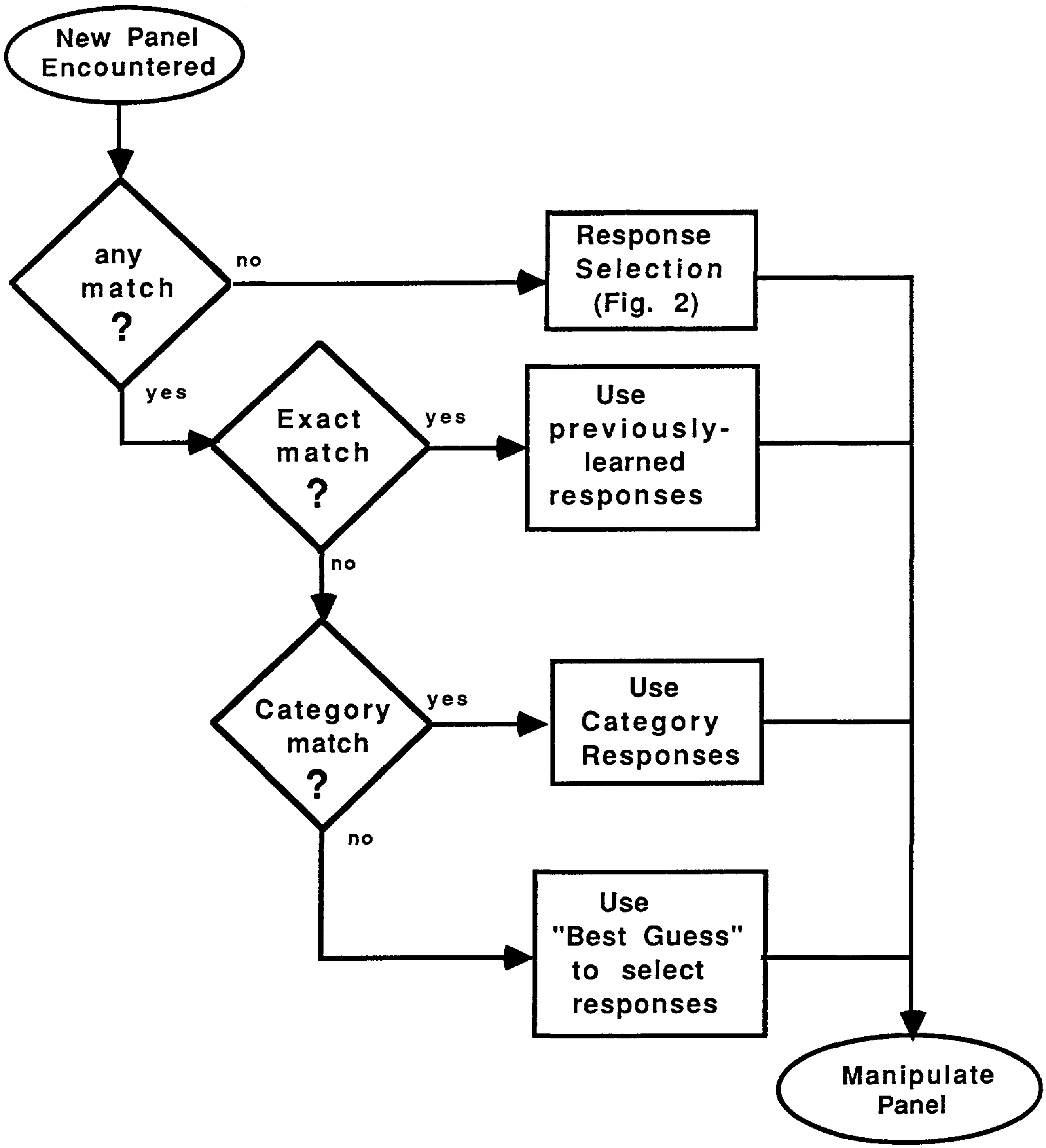

Figure 1. Logic of Hypothesis Generation.

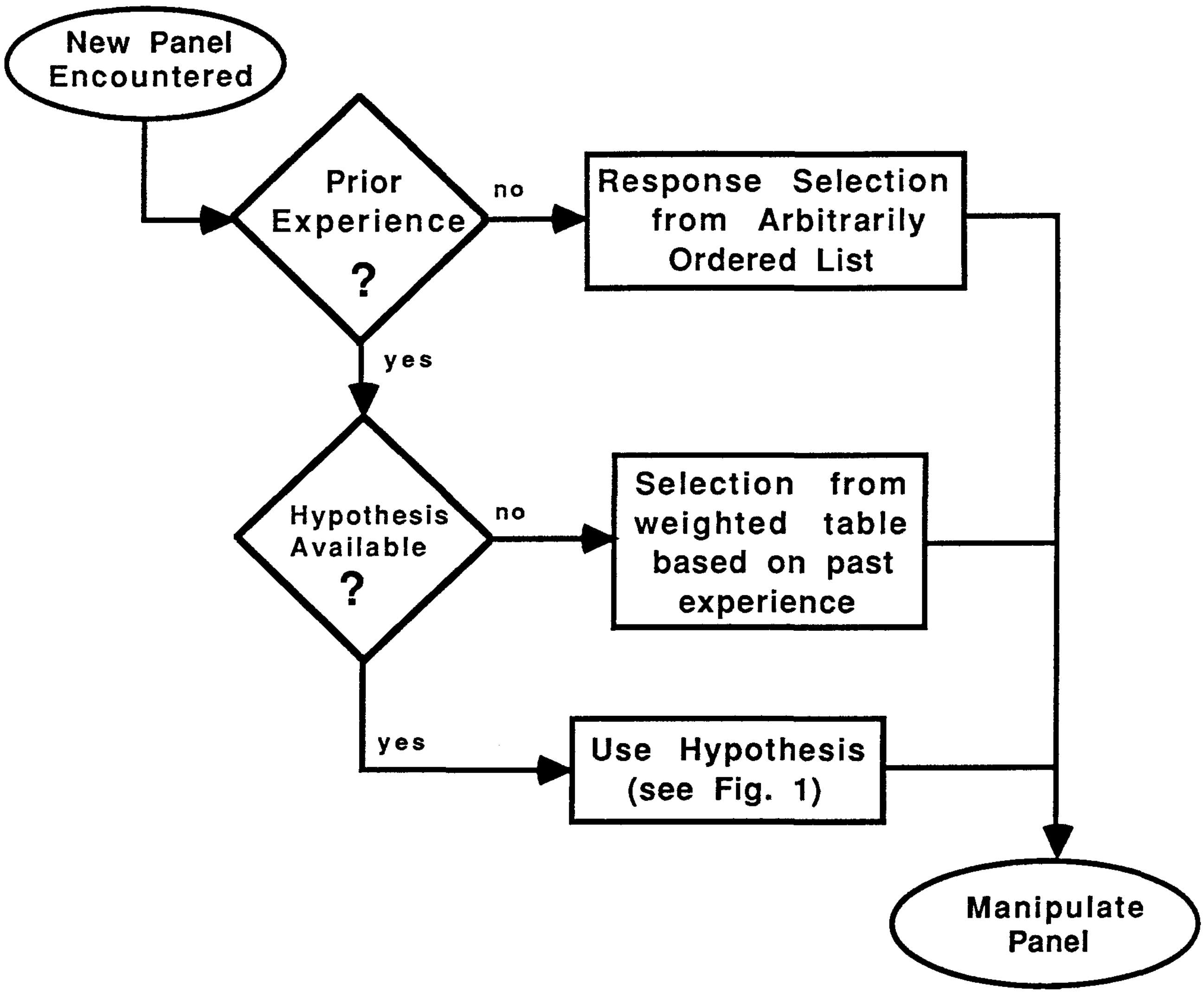

Figure 2. Flow diagram of Response Selection section of the Learning Expert System.

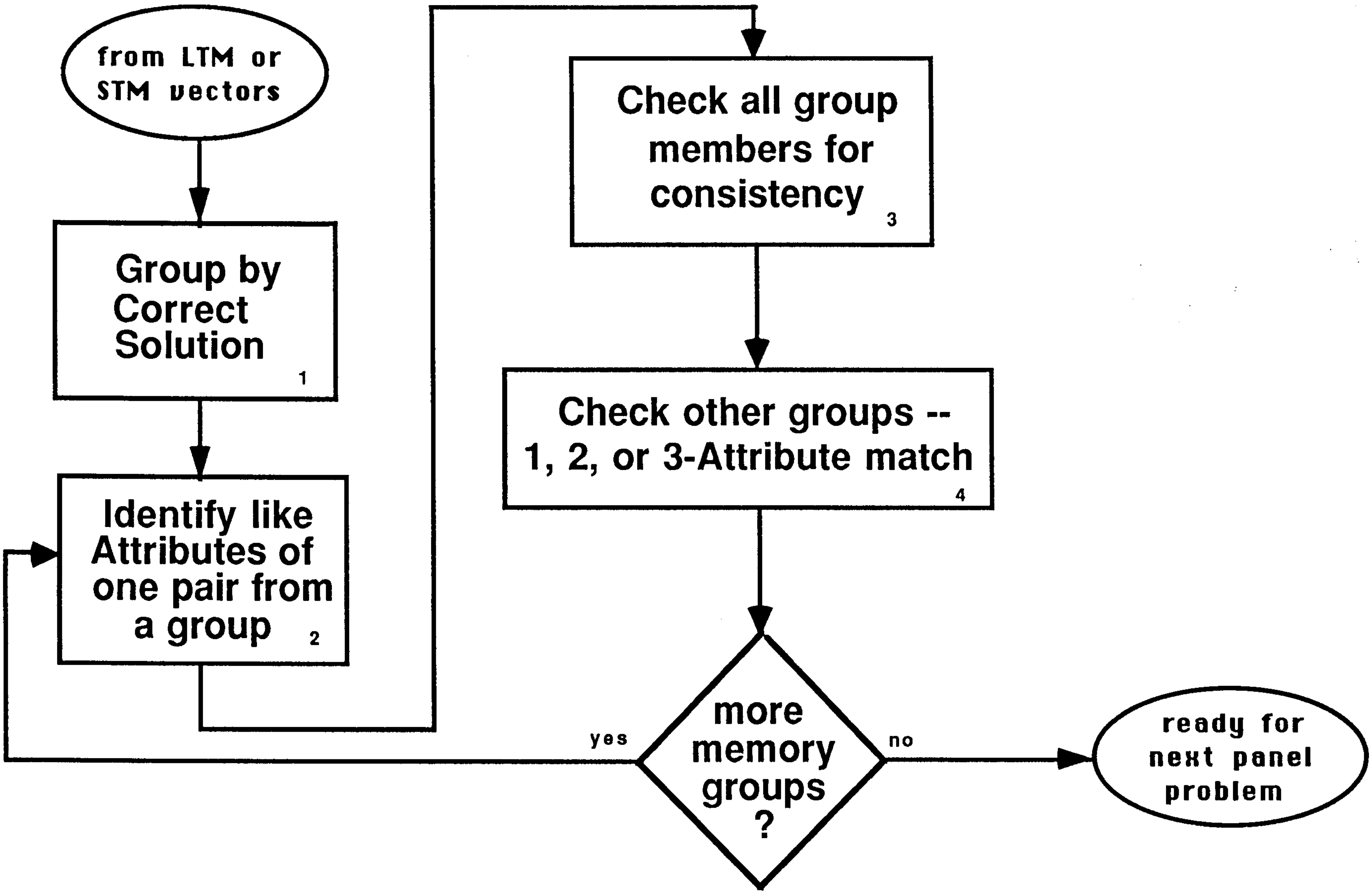

Figure 3. Diagram of the Concept Formation Section.

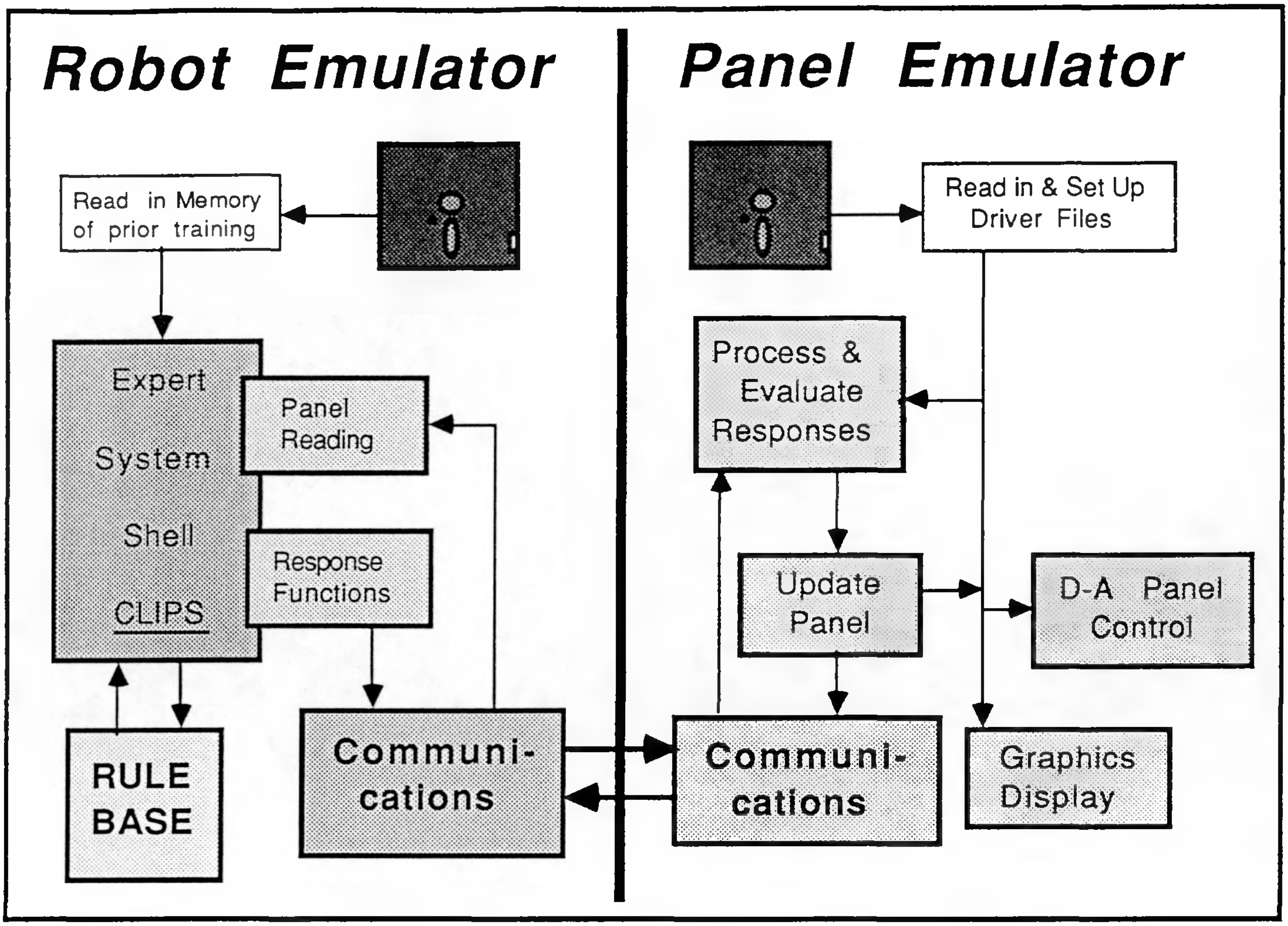

Figure 4. Block diagram of 2-PC simulation system for training a Learning Expert system.

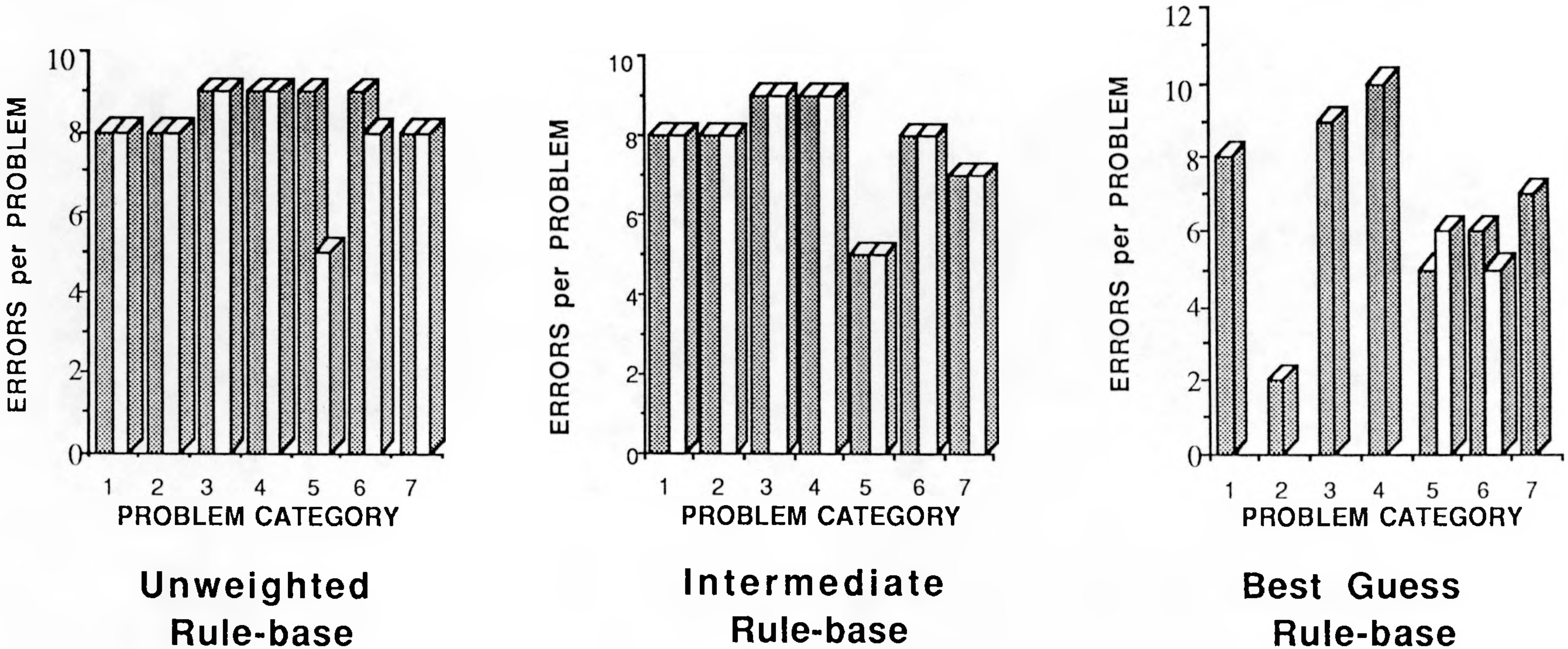

Figure 5. Errors per problem for the three rule-base versions for the 14 problems presented to train the robot on a 7-category scheme.

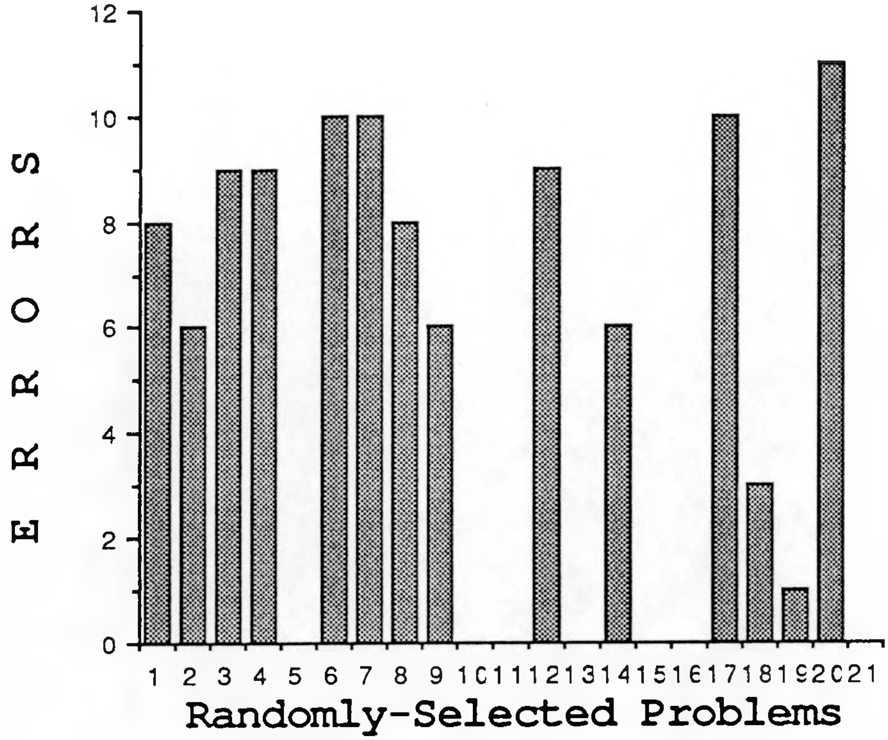

Figure 6. Typical performance on 21 of 81 randomly-selected panel problems.

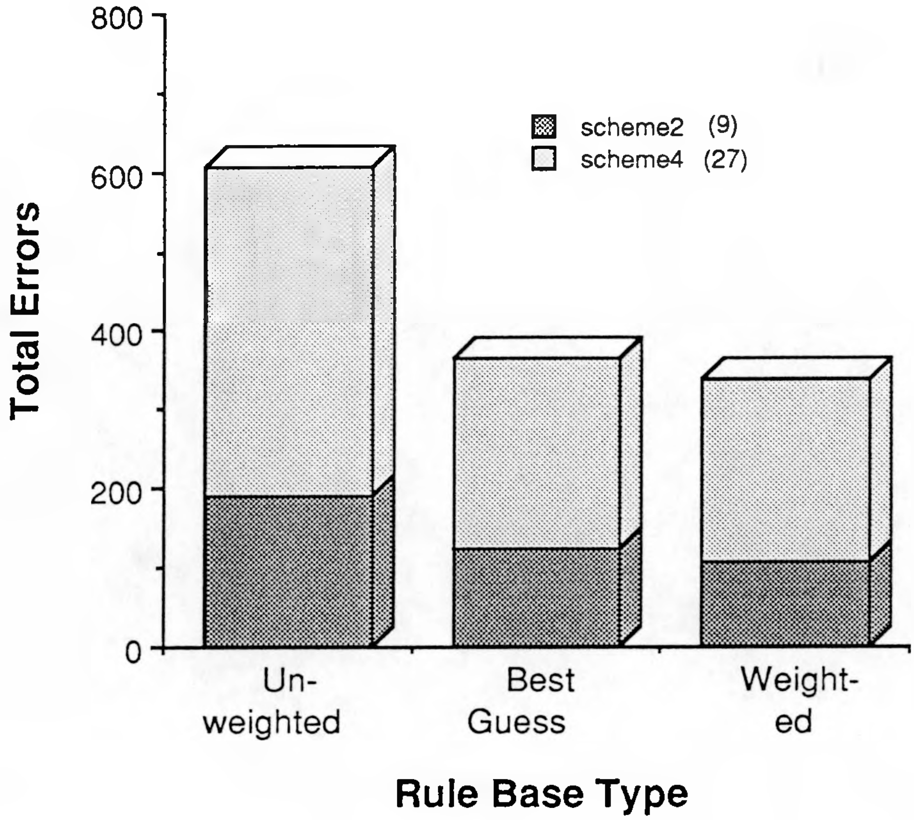

Figure 7. Errors on 81 problems by three versions of the rule-base, for schemes with 3 (gray) or 27 (speckled) categories.