General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

JPL PUBLICATION 77-16

Robot Learning and Error Correction

(NASA-CR-153202) ROBOT LEARNING AND ERROR CORRECTION (Jet Propulsion Lab.) 30 p HC A03/MF A01 CSCL 09B

N77-24834

Unclas

G3/62 29227

JUN 1977
RECEIVED
NASA STI FACILITY
INPUT BRANCH

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91103

JPL PUBLICATION 77-16

Robot Learning and Error Correction

Leonard Friedman

April 15, 1977

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91103

Preface

The work described in this report was performed by the Information Systems Division of the Jet Propulsion Laboratory.

Contents

Introduction 1

Recognizer 3

What Changes as a Result of Learning 6

A Perception Learning Scenario 7

Underlying Principles 15

Error Correction 15

Learning Applied to Error Correction 18

Learning Stimulus — Response Pairs 22

Recognizer Status 23

Summary and Future Directions 23

References 25

Figures

1. The JPL Robot 2

2. A CSA decision net 4

3. Robot semantic net before learning 8

4. Robot outcome net for "grasp" 9

5. Robot outcome net for "move-object" 13

6. Robot outcome net for "ungrasp" 14

7. Error outcome net for "grasp" 19

°. Semantic error net for object-grasp 20

9. Position error normal to finger plane 21

PRECEDING PAGE BLANK NOT FILMED

Abstract

A model of robot learning is described that associates previously unknown perceptions with the sensed known consequences of robot actions. For these actions, both the categories of outcomes and the corresponding sensory patterns are incorporated in a knowledge base by the system designer. Thus the robot is able to predict the outcome of an action and compare the expectation with the experience. New knowledge about what to expect in the world may then be incorporated by the robot in a pre-existing structure whether it detects accordance or discrepancy between a predicted consequence and experience. Errors committed during plan execution are detected by the same type of comparison process and learning may be applied to avoiding the errors. The model is being implemented as a system called RECOGNIZER, and will be incorporated into the existing JPL robot system so that its performance may be tested in real situations.

Descriptive Terms: robot learning, error correction, partial matching, association, recognizable states.

INTRODUCTION

We describe a learning paradigm designed to improve the performance of a robot in a partially unpredictable environment. We will discuss closely related work on error correction, showing how the learning process may be applied to it, and how research in partial matching of patterns can be used to provide the necessary support for the learning process once it has been initiated. The inspiration for the work on learning and error correction reported here has been the JPL Robotics Research Program. A robot system has been under development at JPL for five years and is now fully operational, integrating vision and scene analysis subsystems with both manipulation and locomotion (see Fig. 1). A brief overview of the robot's system organization is given in Thompson's paper on robot navigation (Thompson, T1).

The learning paradigm, which is being implemented in a system called RECOGNIZER, has been described elsewhere with some indications of its application to modeling biological learning (Friedman, F1). Here we are concerned with its interactions with error correction and partial matching. The starting point for both the learning and error correction processes is the *recognizable state*. By *recognizable* we mean two things. First, an internally stored model of the state exists. Second, a process for matching sensory inputs against the internal model also exists.

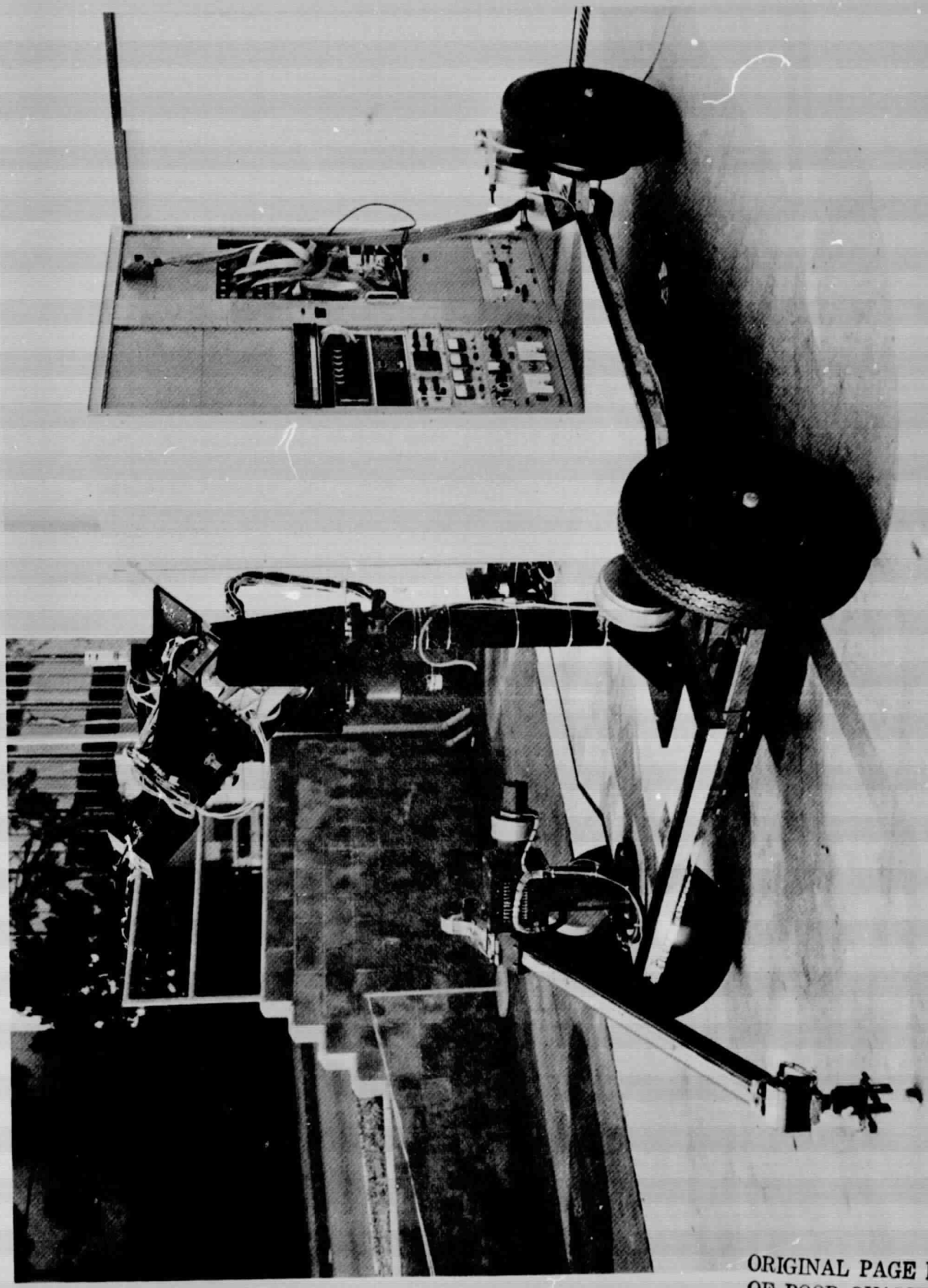

Fig. 1. The JPL Robot

ORIGINAL PAGE IS
OF POOR QUALITY

RECOGNIZER

RECOGNIZER will associate object descriptions or perceptions from the scene analysis system with perceived consequences of robot actions. In order to make clear what is meant by learning in this paper, we will first describe elements of the Common Sense Algorithm (CSA) language in which RECOGNIZER is being programmed. CSA is a high-level language system under development by C. J. Rieger for use in natural language understanding (Rieger, R1, R2, R3).

The JPL robot employs, in addition to various support procedures, a subset of procedures that accomplish useful functions in scene analysis, manipulation and locomotion. A member of this procedure subset is called an "action." A string of actions, called a "plan," can achieve specific goals of a human operator. In order that a plan-synthesizer may be able to construct a plan in RECOGNIZER, knowledge about robot actions is provided by the designer and stored as a CSA form. This form is a triple, linking the name of a robot action and its parameter list with the name of the state it produces via a causal link (R1). The form also includes slots for preconditions or gates. These are states that must be true if the action is to produce its intended effect. For useful plans to be constructed, the uninstantiated algorithms must be selected and instantiated. A decision net for each goal-state performs this function. The selection net for a goal-state is called a "causal" net and consists of nodes, arcs, and the terminal algorithms. A test performed at each node chooses the arc to a successor node and to another test, eventually reaching a terminal algorithm (see Fig. 2). When the CSA plan-synthesizer receives a request to make a goal state true, it traverses the corresponding causal net to a given algorithm, examining that algorithm for gating state conditions not already true. For each of the gating states in turn, the process of traversing

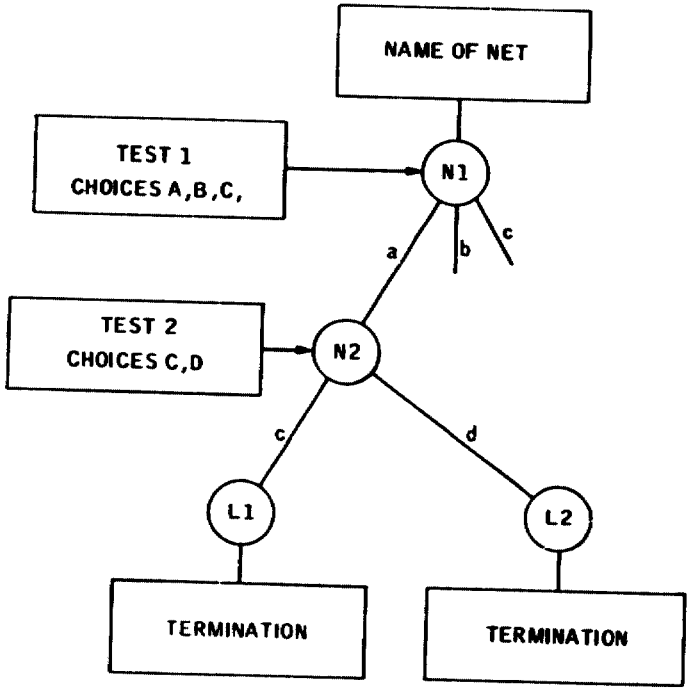

Fig. 2. A CSA decision net

its causal net is repeated till an action to make each gating condition true is found. The synthesizer then links the actions in proper order to make a plan that accomplishes the desired goal state. There is also a generalized "demon" capability provided.

This brief description of the language suffices for our definitions and we can now describe RECOGNIZER itself. RECOGNIZER incorporates a causal net for each action in the robot's repertoire. Other decision nets are also employed in the system. For each robot action, there is an "outcome" net. This is a decision net that terminates with measurable predictions of what may happen as the result of an action. The predictions take the form of more-or-less directly sensed input parameters such as "finger touch sensors 1 and 2 are off" or of higher-level perceptions inferred from these patterns such as "unsupported rock." Still another type of net is the semantic decision net which selects for perception categories based on the descriptions constructed from the sensory input. One semantic net infers useful property categories of objects perceived, another the existence of conditions leading to the commission of errors.

Each semantic net furnishes a corresponding outcome net with the information needed to make a decision about what is the *expected* outcome, selecting from all known outcomes of an action. If such a partnership exists for every robot action, expectation can be compared with experience. (Specific examples of outcome nets and semantic nets are given later.)

The specification of the categories of objects that the robot needs to know and the kinds of error states that it can readily detect requires close study of actual robot experience by the semantic and outcome net designer and an intimate knowledge of robot subsystem design. With this knowledge, he can

specify particular outcome states the robot can measure without knowing the nature of the object or environmental state which will produce that outcome state in advance.

One additional CSA feature facilitates learning. This is the ability to perform "reverse search" of decision nets. The net is normally traversed from top to bottom, with an initial test leading to an arbitrary number of further tests, terminating in some kind of executable statement representation (terminal algorithm) or datum. In CSA, it is possible, after making such a traverse, to start at the termination and retrace the path actually followed in reverse, because the result of each test has been remembered. By arranging a system which plans action strings from a knowledge base of causal nets and which has some expectation of what it will sense as the result of each action it will take, we can relate what is perceived during execution with the anticipated sensed consequences. Reverse search enables us to locate critical branches at which to place the learned perceptions.

WHAT CHANGES AS A RESULT OF LEARNING

Two forms of learning will be discussed, learning how to categorize specific unknown perceptions and acquisition of stimulus-response pairs. In categorizing perceptions the tests resident at a given node of a semantic net are subject to modification. The net structure (number of nodes, the arcs leading to successor nodes and the terminations) remains unchanged. At the start, before learning, most of the nodes will have only default tests; i.e., they will have no templates to match against a pattern perceived externally. When no templates are present that match, the default test points to an arc that is most likely when the robot's world is behaving normally. A succession

of default choices leads to a terminal perception category that is most likely. We make the assumption that the environment is regular enough to justify pre-selection of a normal or default node.

After perception learning takes place, there will be templates at those nodes where there were initially only defaults. When an incoming perception matches such a template, a non-default arc is chosen, leading to a non-normal perception category termination.

During stimulus-response acquisition, the structure of the semantic net is modified to add new terminations as well as new templates at the nodes.

When and how the templates are generated by RECOGNIZER and how they are positioned at the appropriate node are described next.

A PERCEPTION LEARNING SCENARIO

A semantic net before learning is shown in Fig. 3. The net provides for a matching of visual perception patterns and can potentially select for intrinsic object properties that affect manipulator performance. The net shown selects for the properties "heavy," "fragile," "sticky" and "hard." "Hard" is the default termination, and will always be selected as the expected category at the outset. Thus the robot will respond by trying to grasp all objects it is commanded to manipulate.

To initiate a learning experience, the robot may be commanded to "pick up rock 1 and put it in the box." The plan-synthesizer will then generate a string of actions including "analyze scene," "find rock 1," "grasp rock 1" "move-object rock 1" and "ungrasp rock 1." Figure 4 shows an outcome net associated with the action "grasp." An execution monitor looks at each action in the plan stack before it is executed. It then activates the corresponding

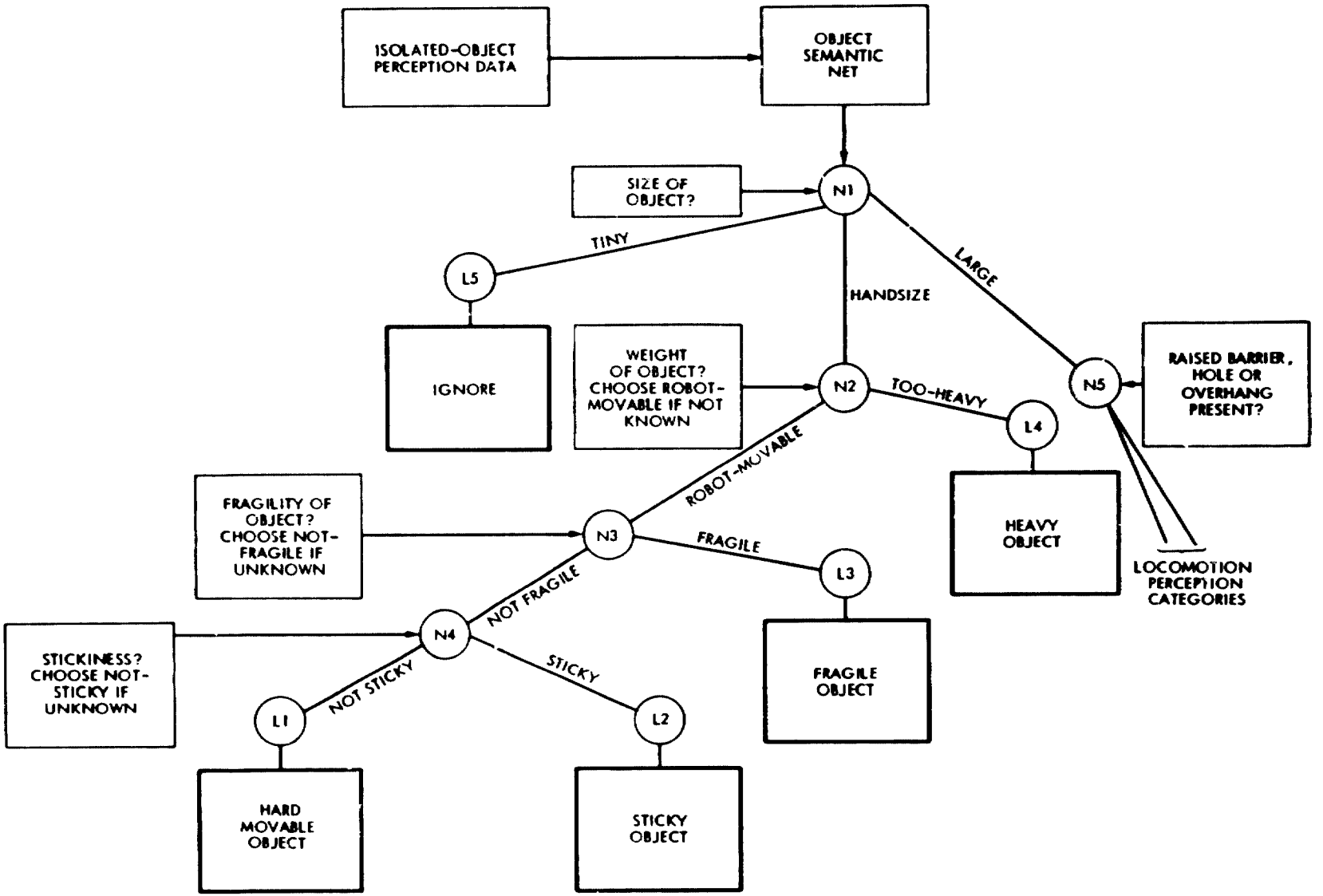

Fig. 3. Robot semantic net before learning

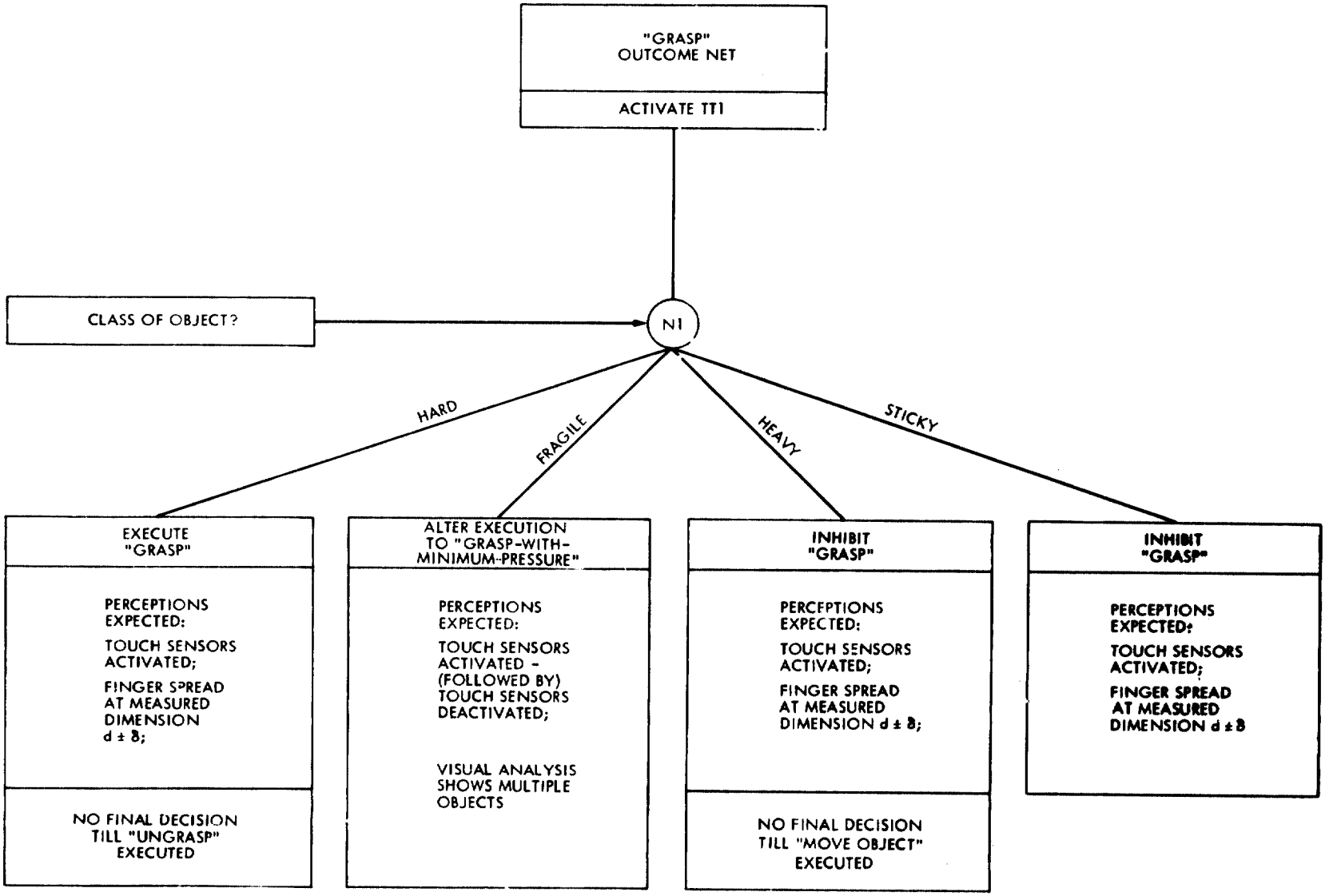

Fig. 4. Robot outcome net for "grasp"

outcome net. In addition to the outcome net, a trigger-tree, TT1, is activated by the monitor. Trigger-trees are CSA constructs and consist of packets of demons (Rieger R3). A demon in TT1 will be on the alert for each combination of sensory inputs shown in the terminations of the outcome net. Thus each termination is a recognizable state. Note that for "grasp" alone the sensors available cannot distinguish between "hard," "sticky" and "heavy." In effect, several categories can be inferred from the action "grasp". These can be disambiguated by subsequent actions.

Before "grasp" is executed, as part of the process of scene analysis and segmentation to find rock 1, the semantic net will make a selection to categorize the object. When the robot is "naive" (before any experience) the semantic net choice will inevitably be "hard object." When the next action is "grasp," its outcome net uses the semantic net selection to make the choice of "hard expected." As "grasp" is executed, the activated demons report to a trigger monitor which compares the demon actually triggered with the expected outcome perceptions. For simplicity, assume that the early experience of the robot will be only with a variety of hard, non-fragile objects.

After each exercise of "grasp," the trigger monitor asks the scene analysis system for its description of the object grasped. The scene analysis system, DABI, designed by Yakimovsky and Cunningham, is working now in the robot system and operates with a library of primitive attributes, specified in advance (Yakimovsky and Cunningham, YC1). An attribute list that is implementable might include shape, size, texture, color pattern, and symmetry. The trigger monitor receives advice from the outcome net in Fig. 4 to wait at least till "ungrasp" for the next step in learning. For both "move-object" and "ungrasp," the outcomes (Figs. 5 and 6) confirm "hard movable object." Therefore

the trigger monitor can proceed. By reverse search, starting at the confirmed termination "hard object," a monitor function climbs the semantic net, placing the description found by scene analysis at each node containing a default till it gets to the highest node in the net. If the experience is repeated, a process capable of determining the common attributes and relations and eliminating differing attributes is employed to revise the test for attributes present in hard objects. Hayes-Roth has described programs for similarity and difference matching (partial matching) between patterns that will do this job (Hayes-Roth, HR1; Hayes-Roth and McDermott, HR-McD1). For the property lists we are talking about here, simple bit operations suffice, performed on binary vectors representing presence or absence of attributes. For more complex relations, his algorithms search the problem space efficiently, and will be employed in RECOGNIZER.

Now the stage is set for learning about exceptional properties such as "fragile." Suppose the robot is commanded to pick up a Christmas tree ornament. It grasps with normal pressure and breaks it. At this point, the trigger monitor discovers that the demon corresponding to a fragile broken object has been triggered and that this is not the expected outcome. Once again it requests the object description from scene analysis, but now starts its reverse search of the semantic net (Fig. 3) from the termination "fragile object." It is looking for the last node common to the path that scene analysis took in the normal direction (to hard object) and the path to the actually experienced termination (fragile object). The monitor can find this node because each time the scene analysis system traverses the semantic net, it leaves an updated marker at each node of the path taken. All the trigger monitor has to do is climb from the termination "fragile object" to N3 in Fig. 2 to find the current path marker. This node is where it will locate its test for "fragile." The

trigger monitor then calls the partial matching process to examine the tests for hardness (non-fragility) at N3. The partial matching process will now seek to find the *differences* between hard and fragile objects by comparing the N3 tests already present with the scene analysis characterization of the ornament as a spherical, smooth, shiny, red object. If a difference set cannot be found, the partial matcher may request more detailed attributes of scene analysis. This is possible because DABI operates with a resource allocation algorithm that controls the time spent and depth of tree search. Thus in a first pass the object might be characterized as spherical. More in-depth analysis would add "a small cylinder sticks out of the sphere." If, on the basis of some predetermined criterion, a distinctively different attribute set description of a fragile object can be found after partial matching a limited number of times, it will be placed at N3, overriding possible similar descriptions for hard objects placed there earlier. Thus the next time the robot is commanded to pick up a similar ornament, its semantic net will choose "fragile object," and its outcome net, by selecting "fragile" expectancies, will find advice for the execution monitor to "grasp with minimum pressure," advice that was not found during its first experience with an ornament until too late.

Similar outcome nets are shown in Figs. 5 and 6 for the actions "move-object" and "ungrasp." These subsequent actions, as already pointed out, serve to disambiguate the properties "too-heavy" and "sticky" from "hard." Once tests for such objects are learned, the predicted expectancies contain advice to inhibit the execution monitor from proceeding further with a planned grasp.

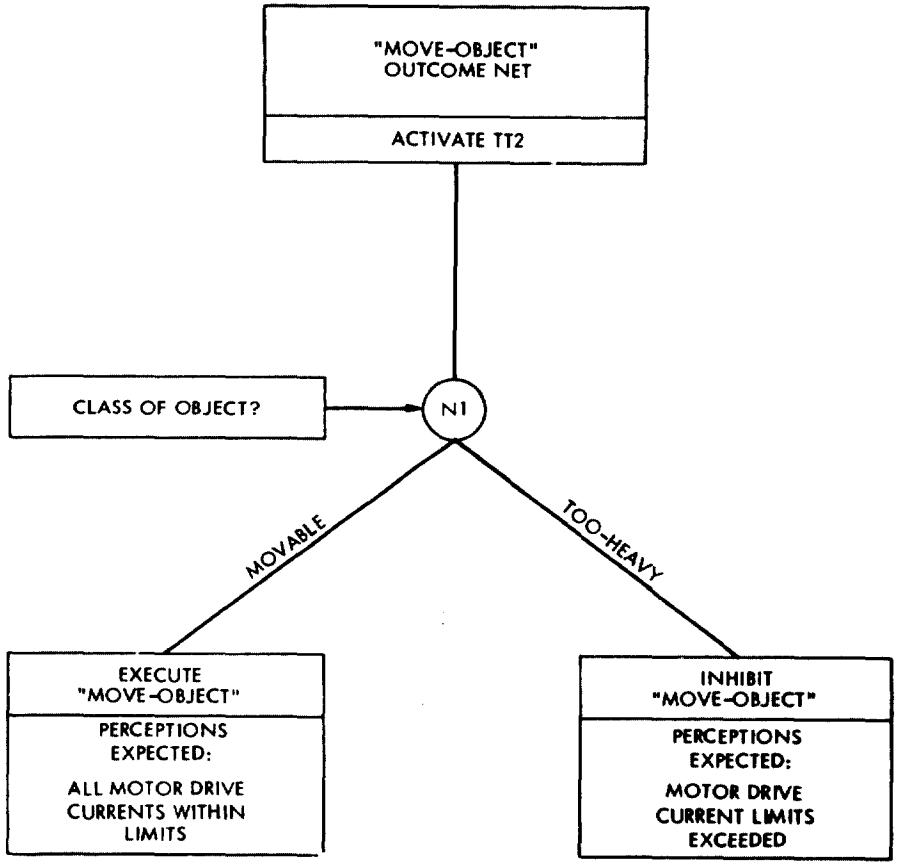

Fig. 5. Robot outcome net for "move-object"

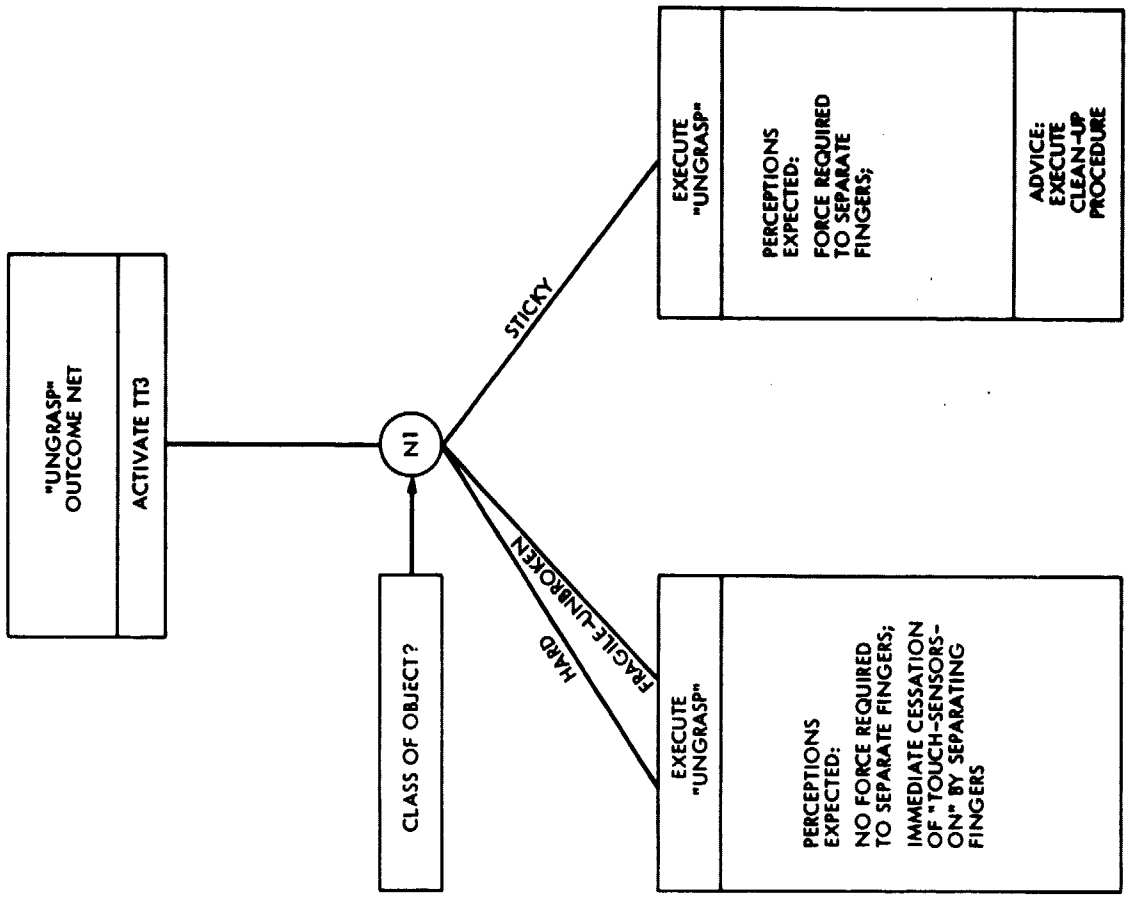

Fig. 6. Robot outcome net for "ungrasp"

UNDERLYING PRINCIPLES

There are several underlying principles of the learning paradigm. First, recognizable state outcomes that are *independent of what is to be learned* are associated with an action (or string of actions). Second, to be useful, the recognizable states must relate to goals of the system such as avoiding danger, correcting errors, locating energy, etc. Third, the recognizable state must be coupled with an action that increases the likelihood that what is to be learned is properly segmented or isolated from the total sensory input. (The designer can only anticipate perception-outcome relationships that are likely, not guaranteed). Thus "grasp" relates to an object whose intrinsic properties (such as weight) may be recognizable via actions (such as move-object) and specific sensory stimuli (such as manipulator motor current overload) divorced from the object's appearance, but the appearance of the object grasped may then become useful information, allowing the machine to avoid further overloads.

ERROR CORRECTION

We turn now to error correction, a subject closely connected to learning, and give an overview of the approach adopted by S. Srinivas (and to be included in RECOGNIZER) for correcting *execution* errors in robot performance (Srinivas, S1). His starting point is also the recognizable state. For each action of the JPL robot he stores a list of possible error states and triggering perceptions actually available in the existing system. For example, the action "move-hand-to-grasp" can be associated with six foreseeable error states. The hand could miss the object to be grasped entirely, left or right fingers could bump into it, etc. Ambiguities similar to those of the recognizable states described in the section on learning also exist here, due to the imperfect knowledge

supplied by the available sensory devices. If in "move-hand-to-grasp" the hand missed entirely, the JPL robot would only know actual hand position (from its angle-sensing pots) relative to desired hand position. It would have to execute the next action, "grasp," to resolve the ambiguity between correct placement and a complete miss.

Srinivas applies two basic strategies for correcting errors after having detected them. These are *failure reason analysis* and *multiple outcome analysis*. In "failure reason analysis" he seeks to determine automatically why the failure occurred by examining the history of actions preceding the failure. When the reason for failure is known, a corrective action can usually be associated with it. The second strategy ignores *why* and seeks to characterize the nature of the error state -- what exactly is the error? Sometimes, simply knowing what is wrong may point to a correction. It appears to be impossible to know in advance which of these strategies (if either) will find a proper course of action to correct the error.

Failure reason analysis is accomplished by synthesizing a tree of causally linked failure reasons and actions. A knowledge base of possible failures for each robot action is provided. These are classified into operational, precondition, information, and constraint errors. Starting with the action at which failure was detected, its associated list of possible failures becomes a candidate for the tree. Some classes of error are causally linked to previous actions. For example, an "incorrect information supplied" reason has the link "incorrectly provided by" which points to a previous action. Before adding a candidate failure reason to the tree, it is pruned, if possible, by a variety of techniques. One method is to examine the sensory manifestations experienced during the performance of the specific action. A manifestation selection net based

on study of that action will point to some of the failure reasons of the candidate list as being more probable than others. Usually the manifestation net will rule out some reasons. The failure tree synthesizer is linked to a trace of previous actions. Once a layer of failure reasons is accepted, those failures causally linked to previous actions provide the actions for the next round of synthesis and pruning. The number of layers added to the tree is limited by the finite trace maintained. If the tree can be pruned enough to narrow the reasons for error to a single cause, a proper course of action for correcting that error is usually determinable in advance and stored with the error.

Multiple outcome analysis seeks to characterize what the error state is by performing additional "inexpensive" tests, when necessary; i.e., causing the robot to execute additional actions for the sole purpose of adding information about the nature of the error committed. This may be needed if the triggering recognizable state indicates an error ambiguously.

If either failure reason analysis or multiple outcome analysis has found a solution pointing to a course of action, the planner goes to work patching in error corrections to the action plan. To achieve the necessary preconditions for the failed action, it may have to undo some actions as well as redo others. Thus, if the failure was associated with "grasp" and the fingers were closed before a failure was detected, they would have to be opened again before retrying "grasp." The resultant "undo" and "redo" steps and new actions are patched into the previous plan and execution is resumed.

LEARNING APPLIED TO ERROR CORRECTION

Error correction refers to the ultimate achievement of a goal state after an initial execution failure. Learning refers to avoiding the failure in subsequent attempts to achieve similar goals. The combined qualities may be called adaptation. RECOGNIZER will incorporate the techniques worked out by Srinivas. Not only are they important in their own right, but they extend the scope of adaptation possible. The learning techniques already described may be applied to the recognizable states classified as errors. Our example will once again center on the action "grasp." Figure 7 shows a second outcome net introduced for the action "grasp" with an additional class of terminal recognizable states. The class previously discussed (Fig. 4) ("hard," "fragile," "sticky," "heavy") do not initiate error correction. The second class ("missed," "position error normal to finger plane [p.e.n.f.p.]," "left finger touching," etc.) when recognized, initiate both learning and error correction processes running in parallel (or simulating parallel processing). Figure 8 introduces a new semantic net, the object-grasp error net. This net is shown before learning and contains a set of tests containing only defaults pointing to the terminal category, "no error." With such a net we can discuss a learning scenario. Suppose that the robot attempts to grasp a rock from above, fails to maintain a briefly attained grasp, and multiple outcome analysis discovers that a "position error normal to the finger plane" exists (Fig. 9). Such a "squeezing out" error occurs frequently. If the robot can categorize shapes such as "wedge-shaped" or "hemispherical," the partial matching process (HR-1) may discover that such shapes positioned between the fingers are often associated with failure. RECOGNIZER will then plant templates for these shapes and for "manipulator position with respect to the object" at the node pointing to p.e.n.f.p.

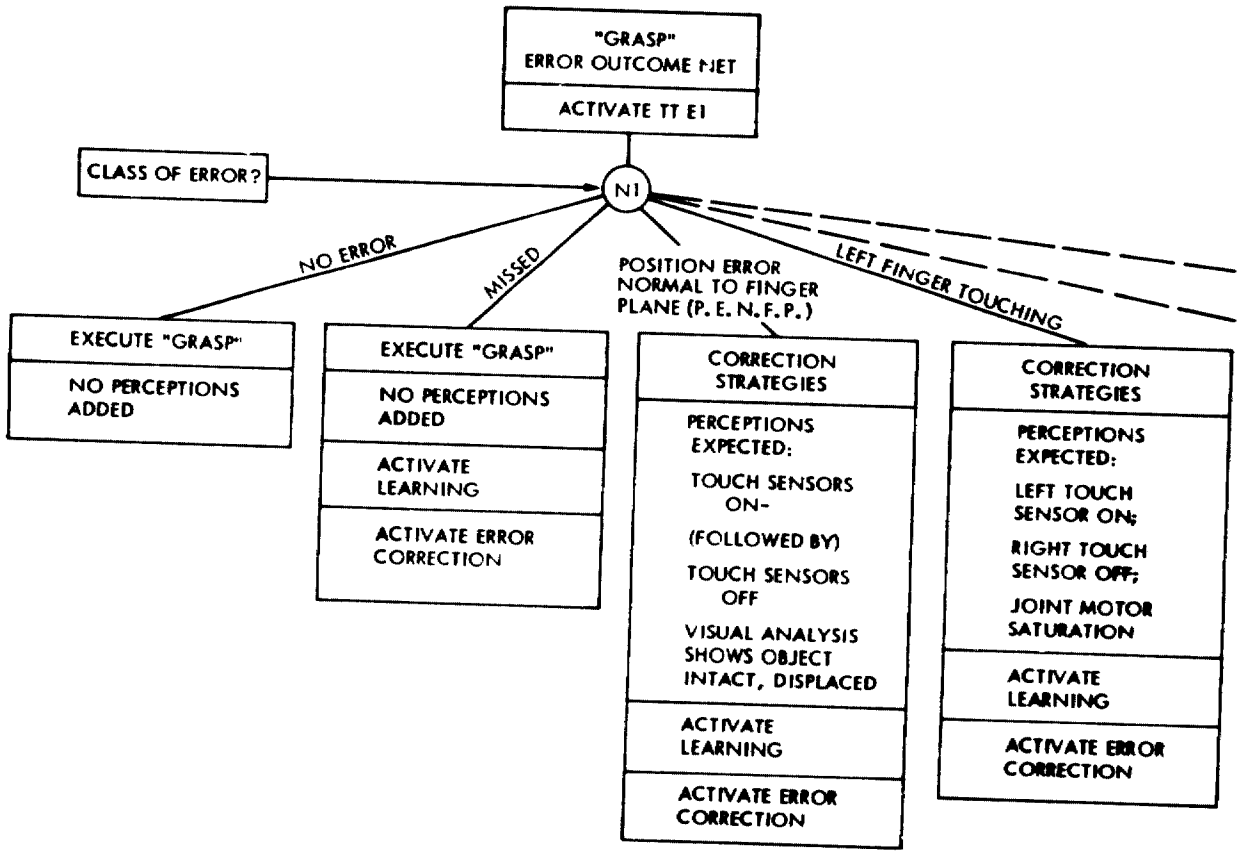

Fig. 7. Error outcome net for "grasp"

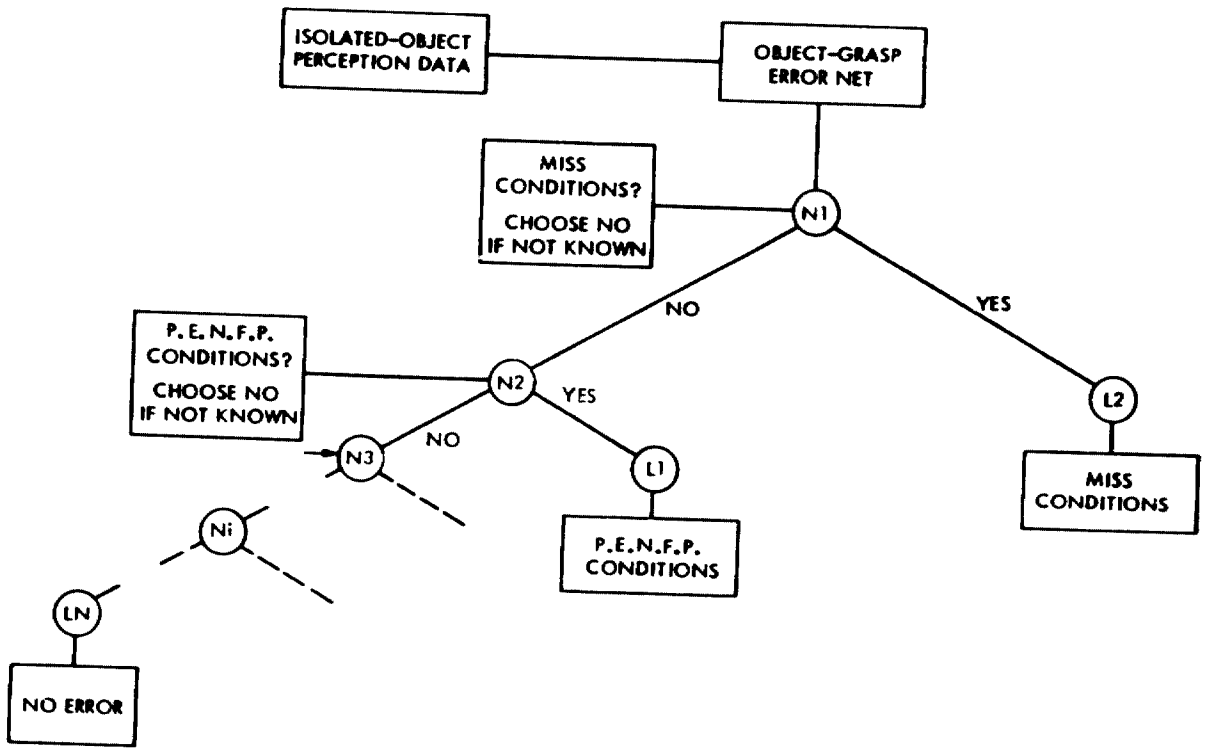

Fig. 8. Semantic error net for object-grasp

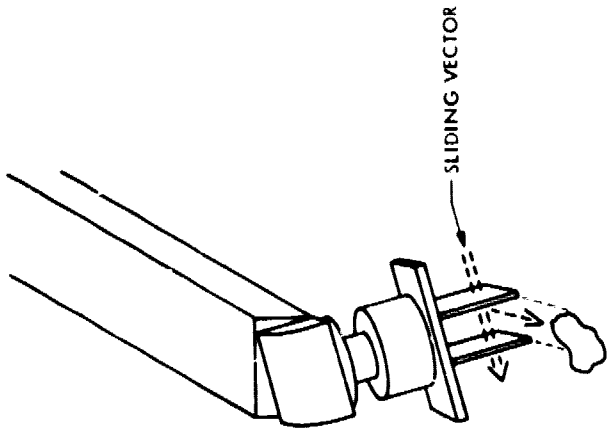

Fig. 9. Position error normal to finger plane (P.E.N.F.P.)

At this stage of learning, RECOGNIZER will expect to commit a p.e.n.f.p. error when it encounters such shapes and will find advice in the "object-grasp" error outcome net on how to modify the robot's actions to correct the error.

LEARNING STIMULUS-RESPONSE PAIRS

Figure 7 indicates where correction strategies are suggested a priori for correcting this type of error. For the p.e.n.f.p. error, in order of increasing motor complexity the correction strategies are:

a) rotate plane of sliding vector

b) change angle of approach of hand from above to side of object

c) provide support underneath object (use a shovel).

For each shape that produces an expectancy of "grasp" failure, the plan-synthesizer can patch in the simplest technique. If it succeeds, an association is established between the successful action and the shape template at node N2 of the object-grasp error net (Fig. 8). The association may be conveniently represented by modifying the structure of the semantic error net, creating a new category "p.e.n.f.p. (technique a)" termination, pointed to by the template associated with it.*

If the simplest technique does not succeed, the error correction process will attempt, in order, the more complex b and c strategies. If they are successful, corresponding new terminations are created in the error net structure. In effect, a stimulus-response structure is created combining an arbitrary perception with a pre-fabricated response. Thus the system may progress from making motor errors of a particular kind to acquiring a kind of motor skill to avoid such failures.

*CSA provides for dynamic restructuring of decision nets.

RECOGNIZER STATUS

A small knowledge base has been provided for exercising the plan-synthesizer, and some demon structures have been implemented, demonstrating the planner's capacity to produce correct robot action strings and for CSA functions to perform reverse search of decision nets. Implementing the elements of RECOGNIZER, and demonstrations of learning and error correction with the robot are scheduled for this year.

SUMMARY AND FUTURE DIRECTIONS

Some features of the RECOGNIZER learning and error correction system design have been described. Although a few general algorithmic principles can be pointed out for the processes denoted here by the terms "learning" and "error correction," the system can be made to work only by studying the actual contexts in which the JPL robot will find itself and incorporating the necessary detailed empirical knowledge in its data base. With such a system made operational, the robot may be able to cope with a less constrained environment than a laboratory.

A recognizable state is one which matches an environmental perception to a stored model. Learning is initiated by such a match. The stored model attributes of recognizable states are independent of what is to be learned. The effect of learning is to enhance the system's ability to predict a relationship between a previously unknown perception of the environment and a semantic category defined by a given recognizable state. When such recognizible states are errors, the system may learn to modify its motor behavior to avoid committing them. The system's ability to accomplish learning depends on finding a description of the perception distinctively different from its previously experienced perceptions.

By studying the actions associated with error states, reasons for failures of the action can be stored in advance, and the knowledge can be used to synthesize failure analyses tailored to the failures actually encountered. We are considering extensions of the error detection and correction capabilities to encompass a wider scope of error. For example, if the robot can link detected conditions like "unsupported" and "above-the-ground" to a model of gravity, it can make better predictions of where to look if it drops a rock. If it is engaged in a complex mechanical assembly and has a model of the completed correct assembly in memory, together with a knowledge of the stage attained, it can detect errors more context dependent than those tied to the general purpose actions of the robot behavior repertoire.

During the coming year we plan to integrate RECOGNIZER into the JPL robot's software system and test its performance in real situations. Such a system when dealing with previously unknown objects will enable the robot to eventually predict intrinsic properties of the objects related to its own goals and so achieve those goals more consistently.

REFERENCES

[F1] Friedman, L., Robot Perception Learning, in Pattern Directed Inference Systems, Waterman, D.A., and Hayes-Roth, F. (Ed.), Academic Press, New York (in press).

[HR1] Hayes-Roth, F., Patterns of Induction and Associated Knowledge Acquisition Algorithms, in Pattern Recognition and Artificial Intelligence, Chen, C.H. (Ed.), Academic Press, New York, 1976.

[HR-McD1] Hayes-Roth, F., and McDermott, J. Knowledge Acquisition From Structural Descriptions, Department of Computer Science report, Carnegie - Mellon University, Pittsburgh, Pa. Feb. 11, 1976.

[R1] Rieger, C., The Commonsense Algorithm as a Basis for Computer Models of Human Memory, Inference, Belief and Contextual Language Comprehension, Proc. Theoretical Issues in Natural Language Processing Workshop, M.I.T., 1975.

[R2] Rieger, C., An Organization of Knowledge for Problem Solving and Language Comprehension, Artificial Intelligence, Vol. 7, No. 2, 1976.

[R3] Rieger, C., Spontaneous Computation in Cognitive Models, TR-459, July 1976, University of Maryland Computer Science Technical Report Series.

[S1] Srinivas, S., Error Recovery in a Robot System, Ph.D. Thesis, C.I.T., Dec. 15, 1976.

[T1] Thompson, A., The Navigation System of the JPL Robot, submitted to Proc. Fifth I.J.C.A.I., MIT, Aug. 22-26, 1977. Also, JPL Publication 77-20, Jet Propulsion Laboratoy, Pasadena, Calif. (to be published).

[YC1] Yakimovsky, Y., and Cunningham, R., DABI - A Data Base for Image Analysis with Nondeterministic Inference Capability, Pattern Recognition and Artificial Intelligence, C. H. Chen (Ed.), Academic Press, Inc., New York, 1976.

NASA - JPL - Coml., L.A., Calif.

REFERENCES

[F1] Friedman, L., Robot Perception Learning, in Pattern Directed Inference Systems, Waterman, D.A., and Hayes-Roth, F. (Ed.), Academic Press, New York (in press).

[HR1] Hayes-Roth, F., Patterns of Induction and Associated Knowledge Acquisition Algorithms, in Pattern Recognition and Artificial Intelligence, Chen, C.H. (Ed.), Academic Press, New York, 1976.

[HR-McD1] Hayes-Roth, F., and McDermott, J. Knowledge Acquisition From Structural Descriptions, Department of Computer Science report, Carnegie - Mellon University, Pittsburgh, Pa. Feb. 11, 1976.

[R1] Rieger, C., The Commonsense Algorithm as a Basis for Computer Models of Human Memory, Inference, Belief and Contextual Language Comprehension, Proc. Theoretical Issues in Natural Language Processing Workshop, M.I.T., 1975.

[R2] Rieger, C., An Organization of Knowledge for Problem Solving and Language Comprehension, Artificial Intelligence, Vol. 7, No. 2, 1976.

[R3] Rieger, C., Spontaneous Computation in Cognitive Models, TR-459, July 1976, University of Maryland Computer Science Technical Report Series.

[S1] Srinivas, S., Error Recovery in a Robot System, Ph.D. Thesis, C.I.T., Dec. 15, 1976.

[T1] Thompson, A., The Navigation System of the JPL Robot, submitted to Proc. Fifth I.J.C.A.I., MIT, Aug. 22-26, 1977. Also, JPL Publication 77-20, Jet Propulsion Laboratoy, Pasadena, Calif. (to be published).

[YC1] Yakimovsky, Y., and Cunningham, R., DABI - A Data Base for Image Analysis with Nondeterministic Inference Capability, Pattern Recognition and Artificial Intelligence, C. H. Chen (Ed.), Academic Press, Inc., New York, 1976.

NASA - JPL - Coml., L.A., Calif.