ROBOT LEARNING AND ERROR CORRECTION

Leonard Friedman
Jet Propulsion Laboratory
Pasadena, California 91103

We describe a learning paradigm designed to improve the performance of a robot in a partially unpredictable environment. The paradigm was suggested by phenomena observed in animal behavior and it models aspects of that behavior. (See FI for details.) An implementation as a working program is under way, intended for incorporation in the now operational JPL robot. Rieger's CSA system is the implementation language and includes a plan synthesizer (RI). A brief overview of the robot's existing system organization is given in Thompson's paper on robot navigation in these proceedings (TI)•

The paradigm includes these components:

(1) A pattern interpretation discrimination net. The terminations of this net are pointers to pre-defined *recognizable* states. A subset of the nodes called default nodes, are initially empty, but a default branch defines a path to a selected termination for this case.
(2) A group of outcome discrimination nets, one for each "action" of the robot's behavior repertoire. Each termination of a given outcome net includes a set of measurable attributes that define a recognizable state. All the states of that net are the foreseeable outcomes of the corresponding action. For example, the outcomes of the action "grasp" may lead to states measured by attribute sets like "touch sensors on" and "finger spread d" for a successful grasp. The outcome net terminations also contain advice to an execution monitor such as "abort action," "alter action parameter."
(3) A packet of demons, one for each outcome net. There is a demon for each termination in the outcome net looking for the corresponding attributes in the sensory input stream.

The learning paradigm depends on an environment already functional in the JPL robot. For example, the scene analysis subsystem is capable of segmenting an input image of the environment into objects with measurable attributes such as size, shape, etc. The paradigm also assumes that plans are available to an execution monitor before the actions are executed via prestored plans or a plan synthesizer.

During plan execution, first the interpretation net is traversed during information gathering, and an interpretive termination is selected. The execution monitor looks ahead to find the next action to be executed and activates the outcome net and demon packet corresponding to that action. The outcome net uses the feature provided by the interpretation net to select an expected outcome. The execution monitor looks in the selected termination for advice to modify the action about to be executed. When that action is executed, the demon triggered by the actual outcome is compared with the expected outcome. The experienced outcome also provides a back pointer to a corresponding termination in the interpretation net. The execution monitor then starts a backward chaining or reverse search operation up the interpretation net starting from this termination. First, the execution monitor requests the attribute set describing the target of the action executed. If the experienced outcome was the expected default choice, the attribute set describing target appearance is placed in all the default nodes in the list pointing to the default path. If the experienced outcome was not the expected choice, the reverse search determines the last default node common to the path leading to the experienced termination and the path to the selected termination. The monitor places the object attribute set at this node in the list pointing in the direction of the experienced termination.

Many things can go wrong with this process:

(1) Some measured attributes may not be germane. The attribute sets retained may be refined by repetition of similar experiences. Depending on whether the expectation is confirmed or refuted by experience, a partial match of the attribute set already resident at the node may seek for a set of similar or differing attributes and eliminate attributes not found to be relevant. (See HR1, FI).
(2) The appearance of the object is not related to the intrinsic property sensed by the outcome experience. Partial matching may eliminate all stored attributes.
(3) Several intrinsic properties that correspond to recognizable states may be present jointly, and the proper course of execution may not be clear.
(4) Objects with similar appearances may have opposing properties (like "dangerous" and "safe"). Depending on the context, a choice in favor of one path or the other may be indicated, although this may permanently exclude some favorable outcomes (FI).
(5) The outcome pattern experienced may not correspond to any activated demon.

As we implement this paradigm and gain experience with its performance, we shall attempt to deal with these weaknesses and the others that no doubt will be encountered.

Error correction here 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. Errors may be detected with outcome nets. An investigation of error correction techniques for execution failures of the robot has been made (SI). These techniques, involving Failure Reason Analysis and Multiple Outcome Analysis will be incorporated as an integral part of the robot learning system.

REFERENCES

[FI] Friedman, L., Robot Perception Learning, in Proc. Pattern Directed Inference Systems Workshop, SigArt Newsletter, June 1977.
[HR1] Hayes-Roth, F., The Role of Partial and Best Matches in Knowledge Systems, Pattern Directed Inference Systems, Academic Press (in press).
[RI] Rieger, C, An Organization of Knowledge for Problem Solving and Language Comprehension, Artificial Intelligence. Vol. 7, No. 2, 1976.
[SI] Srinivas, S., Error Recovery in a Robot System, Ph.D. Thesis, C.I.T., Dec. 15, 1976.
[TI] Thompson, A., The Navigation System of the JPL Robot, Proc. Fifth I.J.C.A.I., MIT, Aug. 22-26, 1977.