LEARNING IN A FREE ROBOT

by

Robert Korn

Computer Sciences Technical Report #296

March 1977

Learning in a Free Robot

Robert Korn

University of Wisconsin

February 1977

Learning in a Free Robot

Robert Korn

University of Wisconsin

Abstract

A "free" robot is one which "pursues happiness" or reward rather than obeying commands. A program for controlling such a robot has been written, and is in the process of being debugged and tested. It is designed to learn to maximize reward while interacting with an environment. The basic mechanisms are extremely general with respect to the nature of the environment, motor abilities, and senses, while fully utilizing known sensory characteristics. Behavior can be directed toward deferred as well as immediate reward and expected reward duration is taken into account. Compounding is used to develop learned hierarchical sensory and motor structures.

Keywords: learning, reward optimization, robots.

The attention of researchers on intelligent robots has been focussed almost exclusively on systems which are "slaves" in the sense that they exist solely for the purpose of obeying commands given to them by human masters. This is understandable since machines are normally developed only for their usefulness to man. But there is another important reason for studying artificial intelligence, and that is to gain a better understanding of natural intelligence. For this purpose it is more instructive to develop a free robot which is able, like humans, to engage in the "pursuit of happiness." The system to be described in this paper is designed to produce motor signals which, based on past experience and recent sensory data, attempt to maximize a reward function. Its actions influence, but do not completely determine, the future states of its environment and consequently its future sensory and reward information. While the research discussed here is concerned with learning of abilities more primative than language and abstract thought, it is felt that these abilities also actively influence, and perhaps determine, the nature of higher level functions.

Considerable research has already been done on robots whose output depends to a large extent on the analysis of sensory (usually visual) data. Most of the work on the larger projects (1, 2, 3) assumed a "slave" type robot which used no learning. These projects also fail to suit our purposes in that they did not interact with their environments by making constant use of a continuous stream of sensory information. This kind of feedback can be extremely useful and is essential in environments which change independently of the robot. A number of smaller systems have been developed in simulated environments (4, 5, 6) which included learning and significant interaction but were overly simple in other ways, such as depending on precise matching of input to stored knowledge in order to learn successfully. The single piece of research which is directly applicable to the problem at hand is Becker's model of "intermediate-level cognition" (7). He outlined a computer program to model the learning of cause and effect relationships, selection of appropriate behavior based on this learning, and development of higher level concepts to facilitate future learning. One problem was that the program itself was never completed and so details were left unresolved and the usefulness of his techniques were never proven. The data representation used, while excellant for expressing time relationships, lacked the ability to express spatial and other relationships. Finally, the handling of reward was not well defined and would have been difficult to include in a working system.

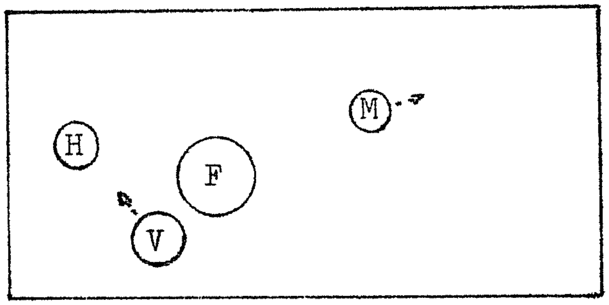
The research to be described here involves the development of a robot control program and a simulated environment for testing it. The majority of the program is independent of any aspect of the environment and could, with suitable hardware, be tested in real world situations. Dependence on known properties of the environment is limited to sensory routines which must be provided to put raw input data into a suitable form and supply other data which is specific to the sensory modality. This data consists of relationships between observed sensory data, a similarity measure between primary sensory quantities, and links matching current items to items in the previous time interval which are probable predecessors. In addition a numerical reward value must be supplied, usually based on current sensory data. At each time interval the learning program produces a vector of real values which represent the degree of activation of its motor channels. It is assumed that these values will influence, but not totally control, the future state of the environment and the resulting sensory data.

The Simulated Environment

The simulated testing environment has been designed to include two characteristics normally missing in real world robot tests: observable motion and activity independent of the robot. In addition the simulation supplies non-trivial sensory data without requiring complex specialized sensory processing.

We attempt to compensate for one important drawback of a simulated environment, dullness, by calling our simulation the "stage world," complete with characters including villains, maidens, and a hero. The hero is the object controlled by the learning system, while maidens are objects pleasant to touch and villains are painful objects. A more realistic, but perhaps duller, way to view the simulated world is as an environment containing a small animal, predators which it finds painful, and prey which it finds pleasureable. The learning program has come to be called HERO, however, in keeping with the "stage" analogy.

The stage world is a two dimensional, rectangular area inhabited by circular creatures who can move around in it. While the number of objects and their characteristics can easily be changed, a typical setup mught include a HERO, a villain, a maiden, and an inanimate object. The movement for each of these is determined by its own control routine. The villain can be set to move toward the HERO while the maiden can be set to move randomly or avoid the HERO. The inanimate object may be specified as pushable by some objects while being a fixed obstacle to others. The HERO is controlled by the learning routine being tested. The positions of objects are updated at fixed intervals small enough so that an effect of continuity is maintained.



Cast

H	HERO
M	Maiden
F	Fixed object
V	Villain

Figure 1. Possible stage world configuration.

If the learning routine is effective, the HERO should in time learn to pursue the maiden while avoiding the villain. The success of the learning routine is measured by the average amount of pleasure received per cycle. An extremely good learning routine would be one which performed as well as a human given the same sensory and motor abilities. A totally unsuccessful learner would be one achieving no more reward than a randomly moving object.

HERO is provided with simplified senses of sight and touch. Since the environment is two dimensional, HERO's visual field is only one dimensional. Each of the objects is assigned a color as a property to be utilized by the vision system. Consider the situation in figure 2. The object marked "H" is a HERO object and the dashed lines show the limits of its vision. Starting at the left edge of the visual field it sees a portion of the west wall which is yellow, then a green region followed by the orange north wall. At the right edge of the visual field a portion of the blue object is observed. A diagram of the raw visual input is shown in figure 3. The visual routine is realistic in that portions of the scene (such as the northwest corner) hidden by nearer objects are not included. The distance between HERO and each observed region is provided along with the other visual information in order to supply depth perception ability.

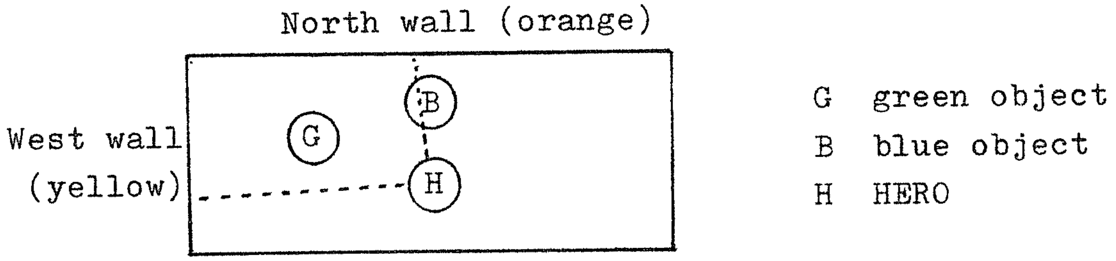


Figure 2. HERO viewing its environment.

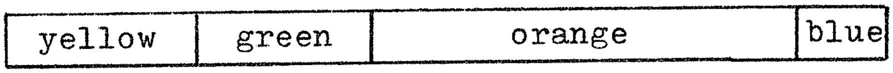


Figure 3. Raw visual data.

Representing perceptual information

The means by which perceptual information is represented is a crucial factor in the design of the rest of a robot system. This is particularly true in the case of the HERO system since an important part of learning is the ability to develop useful new concepts and relationships. Since these may require combining information from different sensory modalities, it is important that we avoid tailoring our representation to a single sense such as vision. Special purpose techniques for improving efficiency when working with particular modalities should be relatively easy to incorporate once the principles for handling a general purpose representation are known. Choosing too abstract a representation can also present problems. Assuming input is an n-tuple of real values is very general but sacrifices the ability to conveniently represent simple relationships such as physical proximity.

The structure of perceptual data for the HERO system was designed to have maximum generality while maintaining the ability to represent relationships in a straightforward way. All data is in the form of directed graphs with labelled nodes.

Special purpose routines must be used to generate graphs for raw sensory input, but the HERO system itself generates the graphs for constructs it develops during the learning process. If, for example, a natural language parser were used to pre-process information for HERO, an input might be represented as shown in figure 4. The symbol ——c is used for the links in the graph instead of arrows in order to reduce the impression that the direction of a link is a limitation on traversal. It is just as easy for HERO to detect that "ate" is the verb referring to "clams" as it is to detect that "clams" is the direct object of "ate." Here the directional characteristics of the links allow us to use "direct object" as a kind of ordered relation between "ate" and "clams." Unordered relations can also be represented, as shown by the way "and" relates "Bill" and "Tom."

"Bill and Tom ate clams"

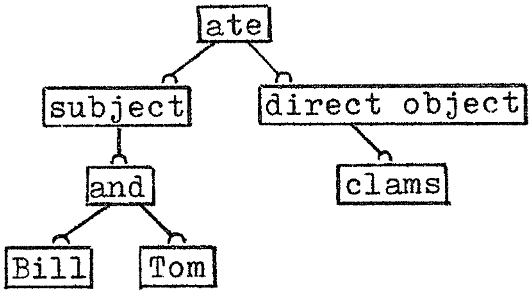


Figure 4. Representation of verbal data.

In order for HERO to be tested in the "stage world" a special routine has been developed to translate raw sensory information into network form. This routine forms a node for each of the regions in the raw visual input. The various properties of each region are each given a node which is linked to the corresponding region node. The properties included are color, position, length, and depth. Figure 5 shows how a portion of a raw visual input would be transformed into graph form. "Position" is the difference in angle between the center of the object

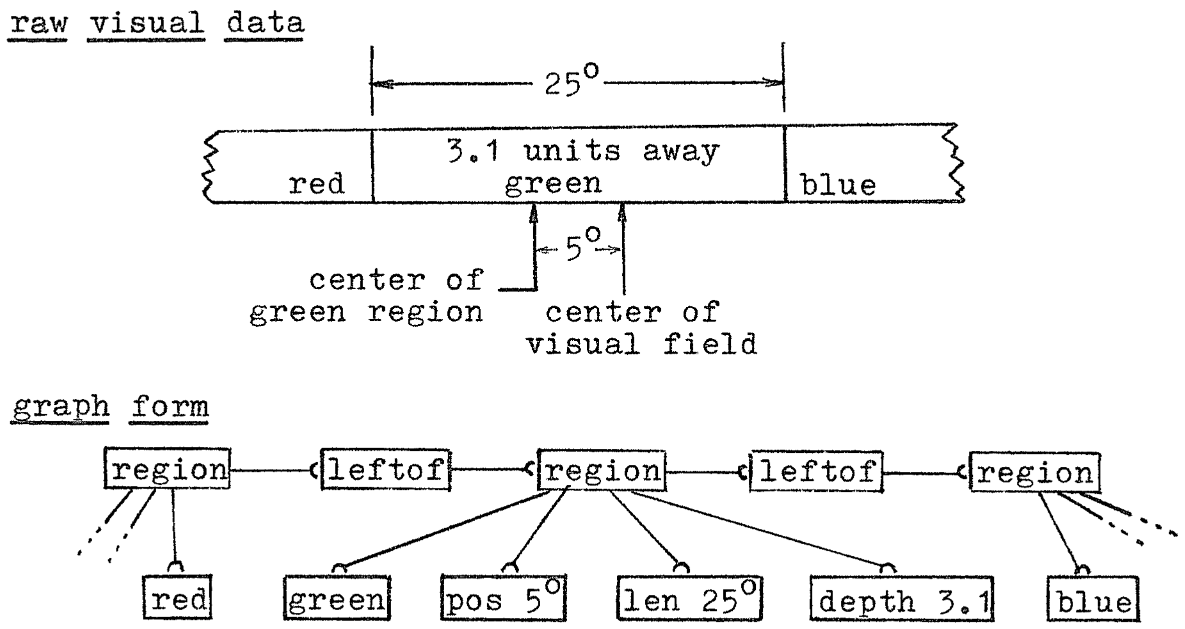


Figure 5. Representing raw visual data in graph form.

observed and the direction HERO is facing. This value may be positive or negative. "Length" is the total angle taken up by the region in the visual field. "Depth" is a measure of the distance between HERO and the observed surface. In order to show adjacency relationships a "leftof" node is used to connect regions. All node labels are numerically coded so that nodes like "pos 25^{o}" and "len 25^{o}" can be easily distinguished.

An advantage of the graph representation is that additional information can be added without disturbing the structure already present. It might be important to know that the green region is closer to HERO than the blue one. This information can be added to the network by including a "closer" node as shown in figure 6. This would eliminate the necessity for frequent comparisons of the depth nodes.

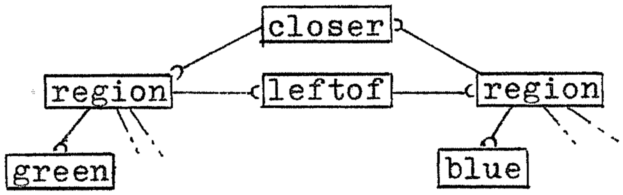


Figure 6. Adding to the graph.

Storing and recognizing situations

There is an extremely simple strategy for choosing behavior which applies to everyday activities from proving theorems to blowing one's nose. Simply stated, that strategy is that we do whatever has worked best for us in similar situations in the past. While a great many problems must be overcome in order to make a strategy like this work well in practice, the generality and simplicity of this idea make it an excellant starting point for designing HERO's learning mechanism.

One problem with this learning technique is that in real life the exact same situation never occurs twice, although most situations are similar enough to past ones for experience to be of use. Therefore instead of looking for an exact match to some previous situation, HERO has an ability to find several of the closest matches and make use of the experience gained from each. Rather than storing complete descriptions of past situations, small samples are used. This saves memory and is realistic in that much of the information sensed at any particular time is usually irrelevant. In this way two situations which may be very different except for a small stored portion may be found to be similar. In addition, provision has been made for nodes in the stored portion to differ somewhat from corresponding nodes in the currently observed network. Functions for measuring substitutability of node labels must be included along with the routines which form networks from sensory data. HERO develops its own substitutability function for node labels generated during the learning process. One respect in which the system does not allow flexibility is in matching network structure. The stored information includes structural data which must be present in the currently observed situation.

Consider a situation in which a green region appears on the left side of HERO's visual field. Using heuristics and chance, HERO chooses to store this part of its visual data in hopes that it may be useful at a later time. A diagram of the information stored appears in figure 7. The asterisks are part

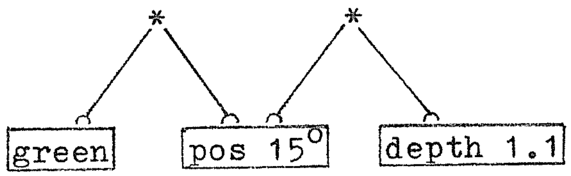


Figure 7. Description with link requirements.

of the linking information and indicate only that two link steps of the appropriate direction are required to get from "green" to "pos 15°" and from "pos 15°" to "depth 1.1."

At a later time the situation shown in figure 8 might occur. The stored description would match the new situation

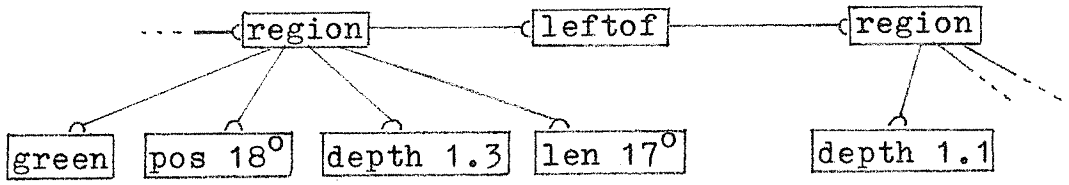


Figure 8. A portion of a situation to be recognized.

fairly well since "pos 18°" is similar to the stored "pos 15°" and "depth 1.3" is similar to "depth 1.1." Note that "depth 1.1" in the adjacent region would not be used in the match because of link structure requirements. An overall strength for the whole match is computed based on the accuracy of matching of the components. Since it will usually be the case that several different matches are made to any particular situation, those with the highest strength are made to have the greatest influence on behavior.

Because of the influence of other matches, the action taken when the "green-left" situation is recognized will often be different from that taken on previous occasions. A record is kept of actions and their success and is used when the situation is recognized again. The most successful actions are positively implied while those leading to poor results are inhibited. Over time the system develops a repertoire of known situations and appropriate responses to each.

Evaluating behavior

The simple "do what worked best in the past" learning technique assumes that we have a good method for determining the degree of success of an action. It is relatively easy to rate actions highly which immediately precede an increase in pleasure, but a properly constructed system should be able to recognize the value of actions which may only lead to pleasure after a significant time delay. In addition the system should be able to handle effects of pleasure duration and pleasure probability in such a way as to maximize its average pleasure level over its lifetime.

The approach taken in the HERO system is to associate an expected future reward value with each situation description stored in memory. The stored value is an estimate of the added total reward due to the presence of that partial situation. Since there will normally be several matches at any point in time, the overall estimate of future reward is the sum of the values associated with the individual matches. This provides a way for actions to be directed toward pleasure which is not immediate. When encountering some pleasurable situation HERO records that the actions immediately preceding the situation were beneficial, and also records that the situations preceding these actions had high expected future reward. Later, actions leading to high expected reward are reinforced. In addition knowledge of expected reward is passed backwards to even earlier situation descriptions.

HERO evaluates actions by using the formula

$$\text{reinforcement}=\text{reward}*\text{time}+\Delta\text{ER}$$

where reinforcement is the value used to compare alternative actions, reward is the actual physical reward following the action and Δ ER is the difference in expected future reward between the beginning and end of the action (in practice the action may continue over several simulation cycles and the calculation is repeated at the end of each cycle). When there is no actual reward, an increase in expected reward is considered beneficial and a decrease is considered harmful. When a rewarding event occurs exactly as expected, the expected future reward drops from the full amount for the event to zero, so

ΔER will have a negative value exactly balancing reward and no reinforcement will take place. If either more or less reward occurs than expected, reinforcement is positive or negative accordingly.

In addition to providing a measure of action value, the quantity we have called reinforcement also tells the system whether its estimates of future reward were too high or too low and by how much. This value is therefore used to modify the future reward values for matches made prior to the reinforcement. The responsibility for the discrepancy in expected reward is distributed among the situation descriptions for these prior matches and averaged with their previous expected reward values. Averaging is used so that probabilistic consequences of known situations will be properly evaluated.

Modifying stored descriptions

When HERO encounters a situation which is unfamiliar, none of its past experiences match very well. Since it must act in some way (no response is assumed to be an action), it is most reasonable that it select the best fit of the various ill-fitting known experiences and act accordingly. If the action resulting from such a match is successful then HERO attempts to generalize the corresponding description by loosening the tolerances for individual components of the description. If, for example, red at position 35° was matched by a description specifying red at position 25° and the outcome was successful, then the tolerances on the 25° specification would be relaxed so the description would apply better to the new case. Although colors can also be generalized to their neighbors in the spectrum, "red" is not in this case because that would not improve the overall match. One component can therefore be altered more than another during the learning of a good set of tolerances. This process also applies to compound properties which, unlike colors and positions, have been generated by the system itself. These are compared by a substitutability function which is also changed to make the match better by increasing the substitutability between the observed and expected properties.

Researchers in learning often avoid making changes when results are successful, but it is felt that in a system like HERO this can be very useful. Even if the action is successful this time, it might not be chosen again in a similar situation because of other incidental factors.

A complimentary process of tightening restrictions takes place when several descriptions apply at once and unsuccessful results indicate that one of the descriptions really should not extend to the given case. The tightening of tolerance and substitutability in this case helps to prevent the mistake from being repeated. As when loosening restrictions, components which matched well will be subject to little change.

In some cases tolerance and substitutability adjustment will not be enough to make a new situation conform to known descriptions. When this occurs a new description is generated and included in HERO's memory. A confidence value is associated with each stored description and is raised and lowered depending on the success of its use. If the confidence becomes very low, the description is deleted to make room for another description in memory. New descriptions are given moderately low confidence ratings so that they can survive a few unlucky results but will not last long if they are ineffective.

Another way in which memory is modified is the storage of actions and their evaluations based on expected reward. If the action actually chosen, which is a function of several matches, is significantly different from that implied by some high strength match, then that action is stored with its ΔER value. Each description develops a list of actions and weights, both positive and negative. Subsequently, the matching of a single description can support and inhibit a variety of actions.

There are some other memory modification techniques not used in the current program which might be added if desired. Description splitting could be used to make two new descriptions from one old one with some critical component used to distinguish between the two. Generalization could involve dropping some component which was apparently unnecessary. Similarly a component might be added to a description which appeared to be already too general.

Compounding

Hierarchical structuring is often the key to avoiding monstrous memory and processing requirements. As an example of this, suppose each letter of the alphabet could be identified by one of eight templates. Then there would be 8^5 or 32768 possible ways a five letter word like "THINK" could be represented. If we needed to store a separate template for each, recognition of the word would be impractical. When using a two level hierarchy in which the letters are recognized separately and then the word is recognized on the basis of the letters, the task is simplified drastically. Eight templates for each of the five letters plus one for the whole word requires the storage of only 41 templates in all.

Humans seem to use similar techniques to recognize and deal with everyday components of their environment. It is important to recognize that information cannot normally be structured into neat, well defined levels such as word and letters. In fact a letter can sometimes only be recognized by the word it is in, rather than the other way around. The factor which provides power is the ability to encapsulate a complex description into a compound which can be used in other descriptions. The HERO system has been designed so that it can make use of the efficiencies of compounding while avoiding a strict level structure.

Actions also should be compounded. Lifting a box requires several serial steps as well as simultaneous actions by different parts of the body. An intelligent system should be able to output high level actions as a unit, but should also be able to control detailed actions individually where necessary. Since almost all actions require sensory feedback, information flow should be possible through short paths involving only simple sensory and motor concepts as well as longer paths involving high level perceptual constructs.

The basic method by which HERO forms compounds is very simple. Once a stored description has reached a sufficiently high confidence level it is allowed to become a component of other descriptions. Up until now, all examples of the "current situation" consisted of nodes in the immediate sensory input. In fact, however, a new node is created for each match made and is included in the "current situation" or STM (short term memory) of HERO. It then may be part of the match of some higher level descriptor which will in turn cause an entry in STM. Each node thus formed is given a label indicating the description from which it was created. Such nodes will be called <u>derived</u> nodes.

There would be little purpose in representing situations in graph form if this were not extended to derived nodes. Each derived node is incorporated into the network by linking it to the nodes which matched its descriptors. Figure 9 shows how this occurs. The direction of the link is decided arbitrarily for each description component when the description is first stored. Using these links, higher level descriptions can specify that derived nodes be linked properly to other parts of the network.

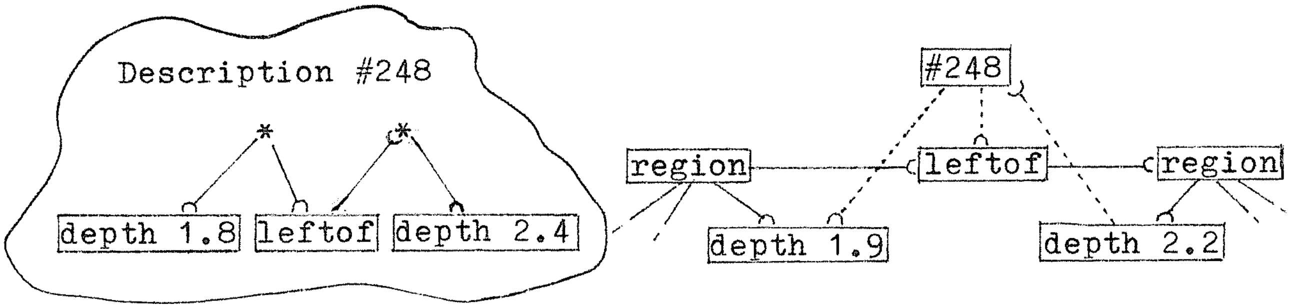


Figure 9. Incorporating a derived node into the STM graph.

The use of derived nodes permits the existence of both direct and indirect pathways between primary sensory data and terminal motor locations. However, if every newly learned description only affects terminal motor locations, then there is no compounding of output. A newly learned description should be able to influence combinations of output once sufficient learning has taken place. It is not necessary to develop an entire separate structure for motor abilities. If a node is several steps away from the terminal motor locations its presence or absence may control considerable motor activity. Thus a newly learned description can influence high level motor activity if its outputs actually influence the presence of such a node. What is in fact done in the HERO system is to permit the node formed on the basis of a description to substitute for nodes which have in the past had a favorable effect on output. Figure 10 shows a schematic view of the network of relationships between input and output and how a newly learned description would influence them.

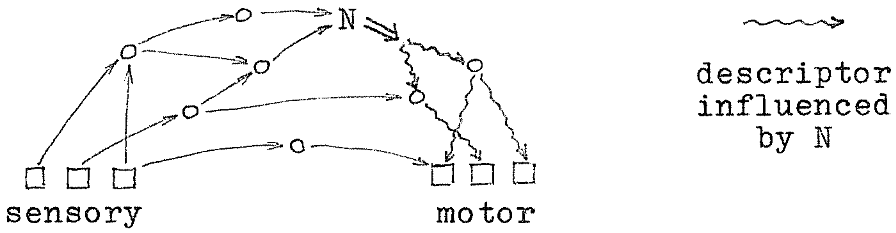


Figure 10. A newly learned node (N) is based on several levels of input and affects several levels of output. Arrows represent description components of the node they point to.

Using this type of network structure it is a significant problem to select the nodes which would make good description components and those which would be suitable to encourage or discourage as actions. In the HERO system a value called ES (execution strength) is calculated to determine the extent to which a node is responsible for the actions actually taken. Primary motor locations which were executed are assigned high ES values which influence ES values for the rest of the network.

The amount of ES assigned to a node is inversely related to its confidence and directly related to the ES values of nodes for which it is a description component. In this way nodes which are still "on trial" and do imply the actions actually taken, get most of the credit or blame for what happens. When an effect on action is desired, this is accomplished by allowing other nodes to substitute for, or inhibit substitution for, nodes with high ES values.

A similar value called *attention* is propagated from sensory nodes to be used in selecting nodes most appropriate for use in new descriptions. This is normally dependent on standard properties peculiar to the sensory modalities such as nearness, central location, and recency of change. Propagation of attention values is cut off by low confidence nodes since these would make poor description components because of the lack of reliability.

Concluding discussion

This paper has presented a broad view of the problems faced in the development of the HERO system and the methods used in solving them. In order to adequately cover issues crucial to an understanding of the overall system, it was necessary to omit a number of other interesting topics. One in particular about which much might have been said is the representation of motion and other changes in STM over time. This, and interactions between it, reward handling, and compounding, are responsible for much of the system's complexity. It is hoped that some of these topics can be covered more fully in future papers.

Two factors which were highly weighted when making design decisions for HERO were generality and simplicity. Whenever possible artificial distinctions were avoided as by not assigning strict hierar~~a~~chical levels or employing separate subsystems for the sensory and motor processes. To the extent that specific information about senses must be provided, the system was modularized so that new senses and changes in old ones could be added without altering other processes. It was attempted to design mechanisms at a level where they could be constantly applied to all thought rather than used only for certain problems and situations. The fact that the various problems of perception, learning, output selection,motivation, and time continuity are all handled in the same system is very important. The requirements for continuous interaction and learning had such an influence on the design of the data structures and other mechanisms that it seems very unlikely that a system of this type could be successful if made up of components developed independently.

The HERO system and the stage world simulation currently exist in the form of a SIMULA program for the Univac 1110 computer. Most of the program has been debugged and tested over short sequences of simulation cycles. Programming has insured that the many details not covered in this report have indeed been worked out in a plausible way. Although limited, the testing done so far has been valuable in showing areas where techniques can be altered to advantage. The program has already evolved considerably and will undoubtedly continue to do so for some time.

The immediate prospects for this project involve testing and modifying the program to maximize useful learning while keeping memory and processing requirements within reason. Over a longer time period it is hoped that underlying theories can be developed for many methods that are now based on judgements of what seems to work. Additional components could be added if they seem necessary for developing the abstract reasoning ability characteristic of human intelligence. The final problem will be the development of a cost effective system for use in the real world. While matching a large description memory to STM is likely to be very expensive on a general purpose serial computer, use of special purpose parallel hardware should be considered if preliminary testing is sufficiently promising.

In the meantime it is hoped that the study of learning while interacting with an environment will provide a useful perspective for understanding both natural and artificial intelligence.

References

1. Feldman, J.A. et al., The Stanford hand-eye project. *Proc. 2d Int. Joint Conf. on Artificial Intell.*, 1971, pp. 521-526.

2. Fikes, R.E., P. Hart, and N.J. Nilsson, Learning and executing generalized robot plans. *Artificial Intelligence*, 1972, pp. 251-288.

3. Winston, P.H., The MIT robot. Machine Intelligence 7, (Melzer, B. and Michie, D., Eds.) Edinburgh: Edinburgh Univ Press, 1972.

4. Uhr, L. and M. Kochen, MIKROKOSMS and robots. *Proc. 1st Int. Joint Conf. on Artificial Intell*., 1969, 541-556.

5. Doran, J.E., Experiments with a pleasure seeking automaton. *Machine Intelligence 3*, (Michie, D., Ed.) Edinburgh: Edinburgh Univ. Press, 1968, pp. 195-216.

6. Jacobs, W. and M. Kiefer, Robot decisions based on maximizing utility, *Proc. 3d Int. Joint Conf on Artificial Intell*., 1973, pp. 402-411.

7. Becker, J.D., *An Information-processing Model of Intermediate-Level Cognition*, Unpubl. Ph.D. Diss., Stanford, 1971.