

Multiresolutional Schemata for Unsupervised Learning of Autonomous Robots for 3D Space Operation

Alberto Lacaze, Michael Meystel, Alex Meystel
Simulation and Modeling Group
IMPAQT Center, Drexel University, Philadelphia, PA
schema@impaqt.drexel.edu

ABSTRACT

This paper describes a novel approach to the development of a learning control system for autonomous space robot (ASR) which presents the ASR as a "baby" -- that is, a system with no a priori knowledge of the world in which it operates, but with behavior acquisition techniques that allows it to build this knowledge from the experiences of actions within a particular environment (we will call it an Astro-baby). The learning techniques are rooted in the recursive algorithm for inductive generation of nested schemata molded from processes of early cognitive development in humans. The algorithm extracts data from the environment and by means of correlation and abduction, it creates schemata that are used for control. This system is robust enough to deal with a constantly changing environment because such changes provoke the creation of new schemata by generalizing from experiences, while still maintaining minimal computational complexity, thanks to the system's multiresolutional nature.

Experimenting with ASR is especially interesting because the rules of input control do not coincide with human intuitions. Actually, we want to see that the simulated device can learn the unexpected schemata from its own experience. Although the traditional approach to autonomous navigation involves off-line path planning with a known world map (such as the potential fields algorithm), in most of the real tasks the environment is not well known because of ever-changing conditions of the assignment absence of gravity, and sophisticated, hard to predict obstacles like components of the space stations, etc. Astro-baby gathers data from its sensors and then by using a schema-discovery system it extracts concepts, forms schemata and creates a quantitative/conceptual semantic network.

When the Astro-baby is first dropped into the space it does not have any experiences and its sensors and actuators are sets that do not have any distinction among its elements. Then, by trial and error, the ASR learns the function of its actuators and sensors; and how to activate them to achieve a the goal given by its creator, or the sub-goals that it finds. In our simulation the initial goal is to minimize the distance to a beacon.

The learning techniques are rooted in a nested hierarchical algorithm molded from processes of early cognitive development in humans. The algorithm extracts data from the environment and by means of correlation, it creates schemata (rules) that are used for control. This system is robust enough to deal with a constantly changing environment because such changes provoke the creation of new schemata using generalization, while still maintaining minimal computational complexity, thanks to the system's multiresolutional nature.

The results of simulation are positive. Astro-baby displays the ability to learn a number of maneuvers.

I. INTRODUCTION

Although the traditional approach to autonomous navigation involves off-line path planning with a known world map (such as the potential fields algorithm shown in [1]), in most of the tasks assigned to autonomous robots, the environment is not well known because of ever-changing conditions of the space, complicated conditions of visibility, and diversified obstacles like trusses, other automated machines, unpredictable objects from other planets. Thus, a system robust enough to cope with changes by means of learning rules about the situation is needed. Motion planning and control for autonomous ground vehicles can be approached based upon substantial human experience of dealing with a diversity of ground vehicles. We believe that 3-D dynamic motion in space requires control rules which are not easily available and are not a part of the intuition of a human designer. Therefore, our intention is to allow the ASR to collect its own rules based upon a system of unsupervised (teacher-independent) conceptual learning.

We have developed a system for early cognition that is capable of extracting concepts from the environment and using them for planning and controlling the ASR. Astro-baby gathers data from its sensors and then by using a rule-discovery system and a concept formatting system it extracts and stores the concepts and schemata to create a quantitative/conceptual semantic network as a system of knowledge representation. The natural growth of the rule-base can be compared with the "subsumption" architecture. However, the subsumption concept does not emphasize the early learning, and is usually designed from prior experience of operation.

Our approach focuses on self-developing knowledge base which starts with a minimal amount of knowledge, which we call "bootstrap-knowledge". The bootstrap knowledge does not include any implicit or explicit information about the world or the robot. It has a minimal set of learning rules which the Astro-baby uses to create a world model, decision-making rules, rules of motion, and rules of perception.

The main idea of our approach is knowledge-base generation by applying generalization recursively to obtain the schemata, or rules of behavior at different levels of resolution from the stored information of experiences properly labeled and organized. During the life of the

unmanned vehicle, these rules are constantly reviewed and updated based on new sensor information and deductions which the Astro-baby makes on the basis of algorithms which are hard-coded in the system ("bootstrap knowledge")

In the beginning, Astro-baby does not have any rule of operation and its sensors and actuators are sets that do not have any distinction among its elements. Then, by trial and error, the space robot learns the function of its actuators and sensors; and how to activate them to achieve a certain goal given by its creator or learned sub-goals. In our simulation the initial goal is to minimize the distance to a beacon (with sensors measuring angle and distance to the beacon with some error) which could be a sunken ship, a lost diver, etc., but because of its learning capabilities, the system's applications could be very broad. Given the goal (expressed as a cost functional) the Astro-baby learns concepts like direction, passageway, or obstacle. If these actuation rules were not to apply in a different environment, it would extract a new set of rules.

The world in our simulation consists of a fully dynamic 3-D environment. We have attempted to incorporate as many variables from the real world as possible, so as to fully test the robustness of the learning algorithm. The environment is constantly changed and no map is given. Astro-baby is a very adaptable system that can both create rules of planning and control and deal with situations that were not envisioned by its creators.

II. LEARNING

Standard Approach

The Artificial Intelligence community has made attempts to write "intelligent" programs, or programs which learn from mistakes, for many decades. Some of the early work is Newell, Shaw, and Simon's General Problem Solver (1956), and Samuel's checkers playing program (1959). Most of these learning systems were built to solve very specific problems of learning. In our Astro-baby, although we take into consideration as many variables from the environment as possible in our simulation, we do not give this knowledge to the learning system. In our research we decided to develop a system which arrives at this knowledge on its own. This cannot be done unless the system is given some initial knowledge [2, 3]. One of the attempts we have made is to find what this minimum initial knowledge should be.

Differences in our approach with other existing approaches are classified below.

A. Drawbacks of Subsumption Architecture

The subsumption architecture is also a multiresolutional one, as is ours. However, in a subsumption architecture, the set of rules of control is predetermined by the designer of the system. This means that the designer must be aware of

all possible situations that the asr will encounter. This precludes the assumption of an open environment and that the system will be able to store all the rules for that open environment and that the designer of the system has all these rules to begin with. This makes applying existing approaches to subsumption for astro-robots impossible. We do not include any heuristic schemata in our system. Instead, we include rules (called "bootstrap knowledge") which help the system to acquire, by itself, through learning, the rules that are given a priori in a subsumption architecture.

B. Multi-Agent versus Centralized Decision-Making

In a multiagent system, the decision-making is decentralized. Thus, it has a set of entities which have their own goals and an arbiter who is in charge of switching or deciding the weight or power of each agent depending on the urgency of the situation. For example, [4] uses a subsumption-based, multiagent approach, generating potential fields of attraction and repulsion in various areas of the map. Some examples of preprogrammed agents are "Follow Object", "Forward Attraction", "Open Space Attraction", "Wall Following". In this approach the environment must be entirely known because of the necessity to determine placement of the potential fields. Moreover, the behavior that the robot should take in front of these potential fields must also be known in order to preprogram these agents.

A centralized control system, in the opinion of its critics, creates a bottleneck by forcing each separate unit of the control system, regardless of the type or resolution of its task, to query one decision maker for instructions. Indeed, this happens if the centralized system is not based upon proper (multiresolutional) task decomposition. The latter not only eliminates the bottleneck but actually reduces the complexity dramatically. In [5] it is proven that a hierarchical system largely reduces the complexity of the computations involved in search.

C. Flat Schemata and Multiresolutional Schemata

An example of learning using centralized decision-making and flat schemata is shown in [6]. When we have a centralized decision-making control system working in a complex environment, the amount of rules that must be dealt with is so large that working in a flat-level is impossible. When we work with centralized learning systems, we must use a multiresolutional configuration to avoid complexity.

The approach of our paper is based upon M. Arbib's theory of motor schema [7] applied to a multiresolutional structure. We believe that high-resolution schemata generalize in such a way as to create a low-resolution level of schemata. This procedure of generalization is recursive

in nature, and is inherent in the learning loop. The reality of computation requires it for complexity reduction.

The Multiresolutional Schemata Approach

A. Theory of Multiresolutional Schemata

There exists a multiplicity of definitions for the idea of "schema" which takes into consideration different aspects of this powerful concept. The concept has existed for centuries, and has recently been applied in the area of neurobiology by [6-8] and others. Schema is a construct which represents an entity related to the areas of perception, knowledge organization, and control.

As far as problems of motion control are concerned, we believe that "schema" should be defined as follows:

Schema is an implication

"situation→action" [9] formulated as an *entity* for a particular i^{th} level of resolution of the world representation. More formally, this statement can be represented as a notation (from [10])

$$\Sigma_i = \{[s_i(t) \rightarrow a_i(t), \rho_i], \rho_i\}, \quad (1)$$

where Σ_i is the "schema",

s_i is the "situation" determined only by a set of the "entity discovered in the set of sensor information at a resolution ρ_i " so that $s_i = \{\pi_i, \kappa_i, \gamma_i\}$,

π_i is the percept: "a set of information delivered from the sensors",

κ_i is the context: "a set of information delivered from the sensors at time $t-\rho_i$ ",

γ_i - is the final goal at a level: "an entity defined by the assignment at a lower resolution level ρ_{i-1} "

a_i - is the action: "an entity defined upon a set of dynamic changes in a position and orientation at a resolution ρ_i "; action is a string of subgoals $\gamma_{sk} (k=1,2,\dots,m; \gamma_{sk} = \gamma_i)$ to be reached before the final goal is achieved, in other words $a_i \rightarrow (\gamma_{s1}, \gamma_{s2}, \dots, \gamma_{sm})$,

ρ_i - is a vector which contains the minimum distinguishable discrete of a spatial dimension or time in the i^{th} level.

The storage of schema is done based upon a concept called semantic network, exemplified in Figure 1.

B. Learning in Multiresolutional Schemata

(1) Bootstrap knowledge

Bootstrap is a minimal set of algorithms which allow us to manipulate a multiresolutional representation of our schemata, which include generalization and task decomposition. The minimal set also includes the rule: "IF <no rule for this situation> THEN <give random signal to actuators>". Other than this, only a "goal" percept and a corresponding cost function are given. This capability and associated learning-related functions are examined in detail below.

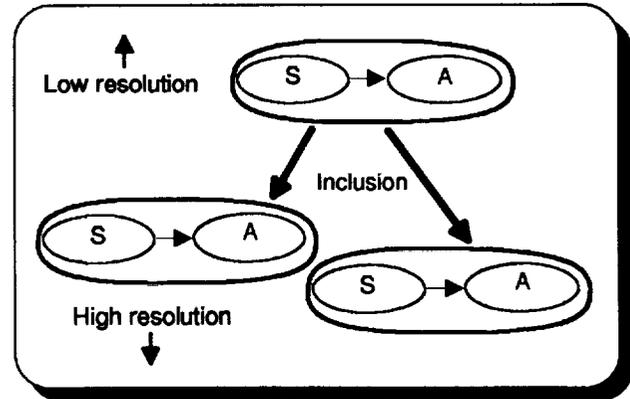


Figure 1. Multiresolutional Schema Representation

(2) Multiresolutional representation

A perfect example of a multiresolutional organization is any linguistic unit. Words form sentences. The sentences form paragraphs, paragraphs form sections, sections form chapters, chapters form articles, and all these articles make books, which also form libraries. Without its multiresolutional hierarchical organization, any book would be a gigantic word. This word would carry all the meaning of all the articles written here. This would create problems not only from an implementation standpoint but also from the point of view of searching through, storing, and communicating. The sentences and paragraphs do not need to be referenced frequently so we do not label them. On the other hand, subsections, sections, and articles carry a label and each of them has different broadness, granularity, or resolution. The title of the book summarizes the content of the book, and is of lower resolution than each of the titles of the articles; the titles of the articles refer to topics that are more specific than the book title. So, we can say that this structure is also nested in the sense that the title of the book includes information about its contents, and so on.

A distinctive property of a multiresolutional organization is the property of "nesting": sets of a particular resolution level are "nested" in a single unit of the lower resolution level. As a result, any multiresolutional representation is a multiple representation of a system at different scales: each level of resolution can represent the same entity with different degree of detail.

Now let's analyze why this multiresolutional nested organization is ever-present:

(a) Search time

Every time we store data -- and in our system we need to do it very often -- this data needs to be retrieved. The Baby Robot stores different percepts and different contexts for further use. These percepts need to be compared to the current percept. Thus, we have to search through the stored percepts. In general, we are interested in performing an

NP-complete procedures without paying for this by any increase in complexity.

It was demonstrated [5] that it is possible to do by repeating the same search several times at different resolution levels: starting with the lowest level (coarse granularity) and performing the search in a large envelope, and ending with a very high resolution space (fine granularity) however, in a very narrow envelope of search. Unlike the search processes (which propagate top-down) the processes of concept generation propagate bottom-up: fine granularity events and entities merge into lower resolution events and entities until the hierarchical tree of percepts and concepts can be assembled. If we store these percepts in a nested multiresolutional manner, our search time will be greatly reduced; it was proven by [5] that searching a nested multiresolutional structure reduces search time.

(b) *Creation of schemata*

Experiences are stored in a form opposite to the form in which the schema is presented (1). Experiences $E_{k,i}$ formulated at the i -th level of resolution for the k -th moment of time are interpreted as our memories about actions $a_i(t_{k,i})$ we performed in response to a particular situation $s_i(t_{k,i})$ and what was the result $s_i(t_k)$ of these actions

$$E_{k,i} = \{[s_i(t_{k,i}), a_i(t_{k,i}), \rho_i \rightarrow s_i(t_k)] g_{k,i}, \rho_i\}, \quad (2)$$

where ρ_i is the value of the increment of "goodness" achieved during the interval of time $\Delta t = t_k - t_{k,i}$.

Experiences are grouped by their goodness in a class of "good experiences". Within this class, a set of subclasses can be created "good experiences at particular situations $\{s_n, n=1,2,\dots,N\}$ ". A generalized statement of experience is declared typical for a particular situation $G[E_{k,i}]$ where G is an operator of generalization. In this paper we will use only the least sophisticated operator of generalization: weighted averaging assuming all weights equal to 1.

Generalized inverted experiences can be considered the basis for transforming them into hypotheses of the future schemata. After a while a set of schemata emerges as a result of inverting classes of similar experiences based upon the value of goodness delivered by a particular action in a particular situation. When creating schemata it is possible, even necessary, to create them to apply a recommended action to an entire class of situations, not just to its members. For example, the Astro-baby might create a rule such as "IF <obstacle visible> THEN <avoid it>". This will include every obstacle that it could sense, and it would not require to create specific rules for every kind of obstacle, every velocity and every direction. This is only possible in a system where lower resolution concepts

include higher resolution ones as the components which are required to accomplish the lower resolution task.

(c) *Task decomposition*

If we create and store schemata in a nested multi-resolutional manner, then actions of a lower resolution level can be decomposed into sub-tasks that are goals for higher levels of resolution. This is done by the virtue of string generation for the higher resolution level in the following manner:

$$a_i(t_{k,i}) \Rightarrow a_{i+1}(t_{i+1,1}), a_{i+1}(t_{i+1,2}), \dots, a_{i+1}(t_{i+1,m}) \quad (3)$$

For example, in the previous given rule, "IF <obstacle visible> THEN <avoid it>", the action "<avoid it>" can be decomposed into "turn right", "orient up", and "slowly accelerate" (Astro-baby creates decomposition which vary with the type of situation). Each one of these actions can again be subdivided until we have a direct command to our actuators.

(3) *Reasoning and Decision Making*

We use only the most fundamental tools of reasoning which are critical for development continuous ("never-ending") processes of learning. Thus, all reasoning is based upon three major operations: a) determining whether a particular entity and/or event are related to a particular class, or not ("issuing the acknowledgment of inclusion"); b) finding an appropriate member of a particular class ("instantiating the class"); and c) forming a new class by determining a group of entities and/or events similar in some respect ("generalization").

The operator of generalization G is the key operator in a multiscale system. It is evoked and utilized to drastically reduce the required amount of computations by allowing to use the "typical" class representative instead of using different particular elements of the class.

For example, generalization is the kernel of the operation that takes schemata in one level of resolution, groups them in order of goodness (given by the cost function), and by means of correlating them it finds features that are in common in the good schemata. These features create schemata of lower resolution that create a new lower level of resolution.

Generalization is recursive in the sense that the highest level of resolution creates a level that is of lower resolution and this lower level creates other level given that sufficient instantiations of this schemata were collected to create clusters form the correlation. In the simulation part of this paper examples of how this generalization works in Astro-baby are given.

III. ASTRO-BABY

One of the possible realizations for Astro-baby is shown in Figure 2. It is possible to demonstrate that this simple configuration is able to provide for all necessary motions. In this paper, we won't concentrate on the subtleties of control for the configuration in Figure 2; we use abstracted translational and rotational vectors of control which should be obtained for any configuration.

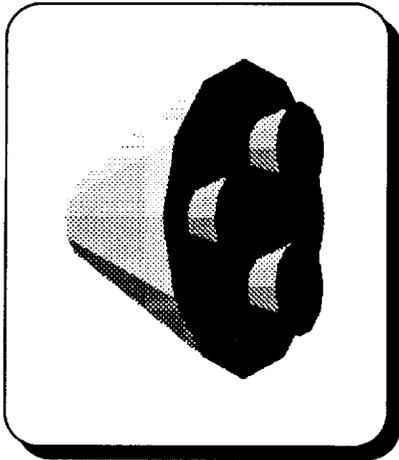


Figure 2. A configuration of Astro-baby

Early learning processes are studied here as applied to the systems which can be represented in a form of six-box-diagram (see Figure 3).

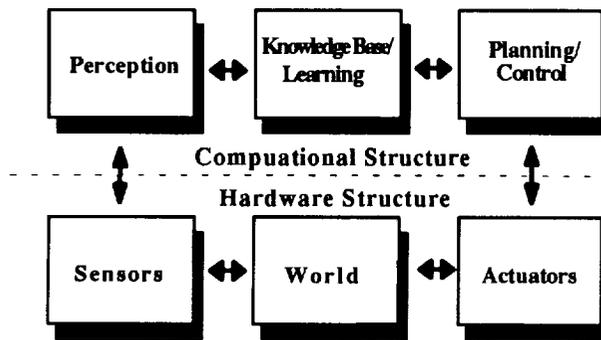


Figure 2. Six-Box-Diagram

The diagram is divided into the Computational Structure and the Hardware Structure which are mapped into another. The Hardware Structure is composed into three blocks: sensors, world, and actuators which are simulated by our program in order to be able to test BR. The other three boxes constitute the structure of intelligence and include: Perception, Knowledge Base and Planning/Control, are the basic components of BR or any control system for that matter. Perception receives the

signals coming from the sensors, quantifies them, encodes them into a language suitable for storage and manipulation, and organizes them.

The Knowledge Bas module receives the encoded sensor information (percepts), puts it into correspondence with the rest of previously stored knowledge and finds relationships (rules) between the actions performed and the concepts perceived. Finally, the Planning/Control ("decision making") module uses all available information and the decision making mechanism to find the command sequence for the actuators.

It was demonstrated that the systems which can be represented by six-box-diagrams has to be equipped by at least two modalities of sensing and have at least two degrees of freedom in their actuation.

In a multiresolutional system, the six-box-diagram is becoming multiresolutional too. Thus it forms a structure of loops which can be called a multiresolutional nested structure (see [11]). In this structure, each lower resolution loop includes generalized activities of the adjacent higher resolution loop. In this paper we will consider a single loop but the results of reasoning can always be expanded to other loops.

The setting for sensing part is easily understood from Figure 3.

Sensors

The following sensors are given to Astro-baby:

A. Distance, Angle to Goal

In our simulation, we simulate real-world sensors by introducing errors. The angles to goal are expressed as Euler angles between the local axes of the ASR and the imaginary vector pointing towards the goal. Some error is introduced in distance to goal.

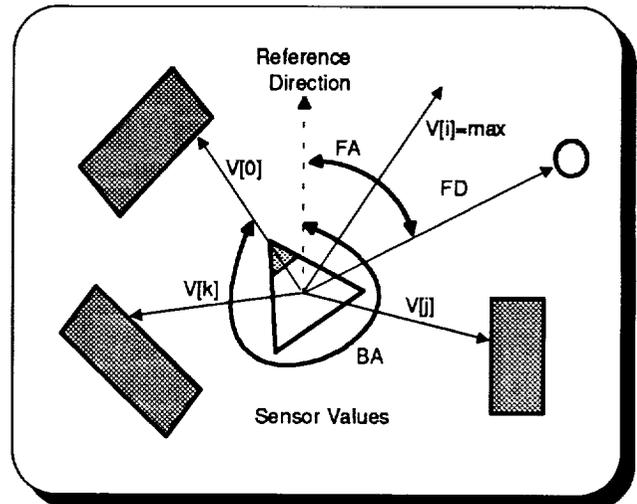


Figure 3. Sensing the position and orientation

Figure 6. Random commands

These random commands generate random motion which is shown in Figure 7. As a result we have a change in the environment and a change in the goodness (change in the distance to goal divided by the step size). So, when the next percept is coming, it has the previous percept, change in goodness, and a context, but it still does not have a rule. But we have the following expression: previous

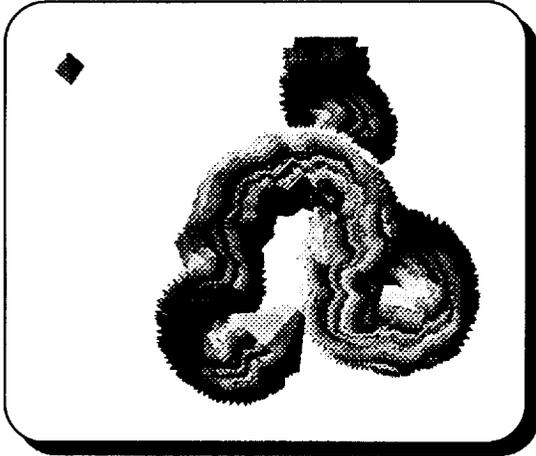


Figure 7. Random movements

percept, action, percept(now) and change in goodness for this action. Thus, we can create, via abduction, hypotheses (schemata that do not have enough statistical data collected either to become full Schemata or to be rejected, we call "Baby Schemata"). The following is a "Baby Schema": IF ((percept(n-1)) && (percept(n)) && (delta(goodness) is desired)) THEN action. The desirable goodness is given by the user as a threshold.

This Baby Schema will probably apply only in very selected situations and as a matter of fact they might give a different goodness in the same situation (because of dynamics) and its goodness could be very low (i.e. going away from the goal). But after we go through this process several times, we have a set of "Baby" schemata that cover some situations. If we have two or more Baby Schemata for the same situation, then the schema with better goodness is applied.

We can see in Figure 8 that this baby schemata causes Astro-baby to "spiral" towards the goal. The use of the baby schemata by Astro-baby improves its operation and at the same time it helps to collect more data of "good" baby schema. Then the generalization process starts working. Baby Schemata are ordered by goodness, and a correlation "engine" tries to find similarities among the baby schemata. First it tries to see if some of the values have been kept constant (within a fuzzy region), then it checks if

the bad baby schemata also have this quality. If not it decides that this is a good characteristic in this class. In the case of the Astro-baby the Euler angle between the nose of the sub and the goal are

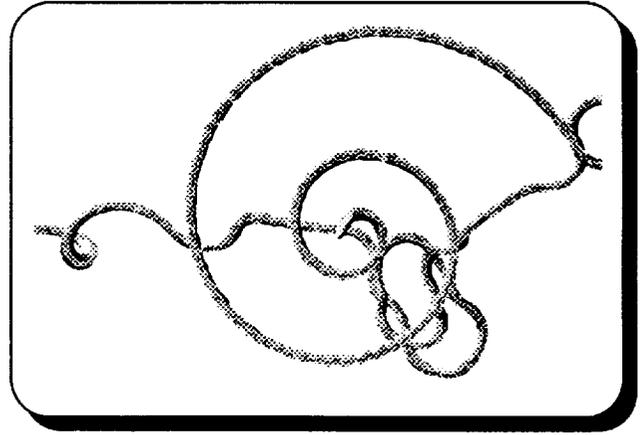


Figure 8. Testing Baby Schemata

very small in all the good schemata, so it creates a new low resolution schema that could be understood as:

$$\text{if } \langle \text{empty} \rangle \text{ and } \text{goal}_1 \text{ are required then} \quad (4)$$

$$\text{make } \text{sensor}_n = 0$$

Where sensor_n is the Euler angle between the nose of the sub and the goal, and goal is minimize delta distance to goal/ delta step. The reason that it puts an $\langle \text{empty} \rangle$ in the Percept and Context parts of the situation is that it could not find any relationships between them in the good situations. When it will encounter obstacles, this part of the schema will not be empty.

Other relationships that we check if they were within a fuzzy boundary are the following: addition, subtraction, multiplication and division of two sensors and deltas of individual sensors (giving Astro-baby the ability to derive). These other relationships could also create schemata if they were good characteristics.

At this point Astro-baby has two levels of resolution, thus our goal (minimize delta distance to goal/ delta step) is passed to the lower resolution level. And this lower resolution level passes to the higher resolution level "make $\text{sensor}_n = 0$ ". The higher resolution level does not know how to do this, so it starts again to give random commands collecting them in baby schemata. But these Baby Schemata are judged with the new cost function. After a few trials Astro-baby creates some schemata that perform a Bang-Bang control on the Astro-baby (see Figure 9) trying to point at all times the nose of the sub towards the goal. In the traces of the tail of Astro-baby can be seen clear marks of this kind of control. The oscillations are big because it

does not have enough data to apply the exact amount of control needed, thus is overshooting.

Once some schemata are formed, new schemata are created that are very similar to the ones already created. These new schemata are used to quantitatively improve the previous ones. For example when schema (2) is formed, $sensor_n$ is not exactly zero but a small number; this number is refined every time the same schema is encountered. Thus the overshoot that we see in Figure 6 will become smaller and smaller.

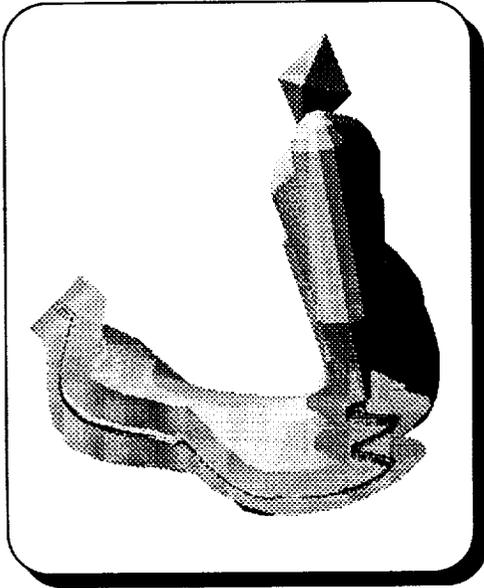


Figure 9. Bang-bang Control

The process of learning does not stop here; new levels of lower resolution appear when obstacles (of any kind: currents, low visibility, etc) are included in its world. The more the variety of circumstance which Astro-baby encounters, the more complex its own control system becomes and the richer its world representation becomes.

LEARNING CURVES

The following experiment was performed:

- a) all knowledge was deleted from the database (except bootstrap)
- b) the vehicle was set in a random position in the screen.
- c) the goal was set in a random position.
- d) when vehicle achieves the goal then go to b)

The learning curves were built by calculating the Euclidean distance between the vehicle and the goal in the initial position and dividing it by the number of steps (time) used to achieve it. The second graph shows the number of schemata versus time.

Case 1

Figure 10a and 10b show the case where no initial random moves were assigned before allowing generalization of schemata. It is possible to see that the learning curve is not very stable, although the performance of the submarine improves, in some trials it has to perform several new random movements to be able to generalize rules that it does not have. It is also possible to see that the number of schemata levels up, the reason for this is that since the simulation is a closed environment, the set of rules that it found is sufficient for its operation.

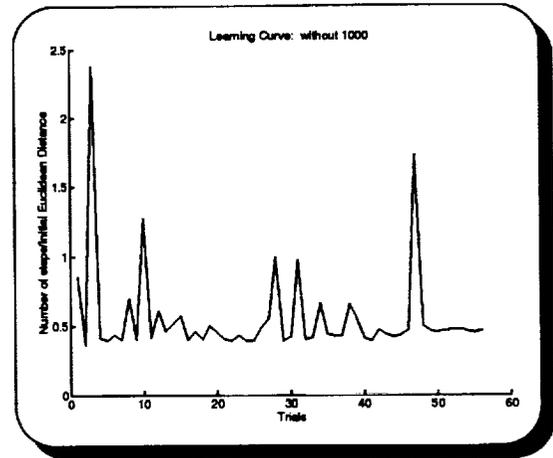


Figure 10a: Learning Curve

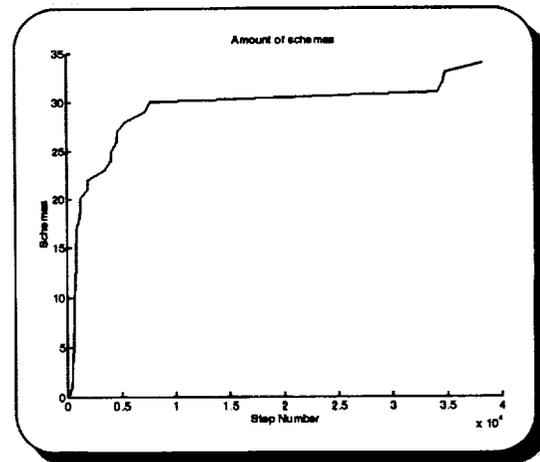


Figure 10b: Number of Schemata

Case 2

Figure 11a and 11b show the case where 1000 random moves were assigned before allowing any generalization of schemata. It is possible to see that the learning curve is a lot more consistent. It is also shown by this curve that the number of schemata found increases faster than in the previous case.

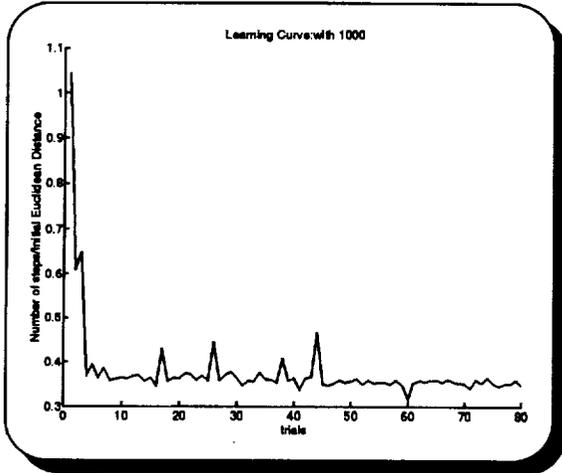


Figure 11a: Learning Curve

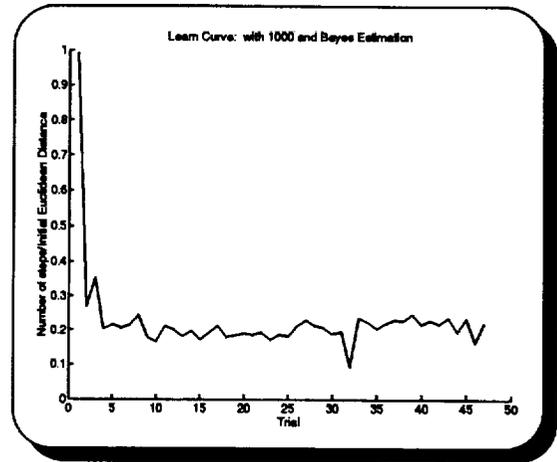


Figure 12a: Learning Curve

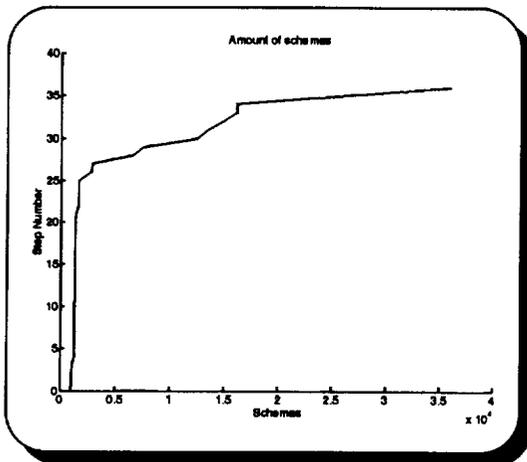


Figure 11b: Number of Schemata

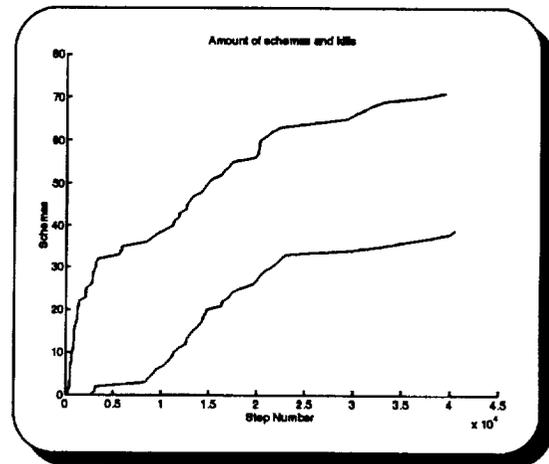


Figure 12b: Number of Schemata and Kills

Case 3

Figure 12a and 12b show also a case where 1000 random moves were assigned before allowing any generalization of schemata and once a Bayes estimator is used to rank the performance of the found schemata. If the Bayes estimator is low then the rule is eliminated. It is possible to see that the learning curve sinks lower than in the previous 2 cases. This is interpreted as the vehicle improving its performance, thus, achieving more distance per number of steps. Figure 12b shows two curves; the one on the top represents the total number of schemata that were created and the one on the bottom shows the total number of schemata that were eliminated using a Bayes estimator. It is possible to see that the number of schemata remains constant.

MAIN RESULT

The early learning process explored in this paper has demonstrated the following sequence of stages:

1. After the random sequence is completed, the learning structure determines that the way to achieve the goal is to minimize (null) the Euler angles. Now this result is considered to be the new goal of operation.
2. As the new goal is pursued, the system learns that it can be achieved by bang-bang control (or variable structure control). The system assigns bang-bang control objectives and they become a new goal.
3. The results presented in positions 1 and 2 can be considered a formation of the low and high resolution levels. If the control objectives of the bang-bang control are considered to be a new goal, the next (the highest) resolution level is formed where the system learns how to provide the oscillation free motion.
4. These stages of the learning process constitute a multiscale system of dealing with experiences and creating a rule based controller. Our conjecture is that the process of

the ("never-ending") learning will continue in the multiresolutional fashion demonstrated above.

5. The bootstrap-knowledge set confirmed to be conducive of the ("never-ending") learning process.

CONCLUSIONS

1. A structure of learning mechanisms for an ASR has been described, based upon a theory of multi-resolutional schemata.

2. A system of simulation has been constructed which allows for testing the process of early learning.

3. The following observations have been made:

Astro-baby is a very adaptable learning system. Adaptable in two senses:

a) when it is installed in a robot it can deal with different kinds of situations and incorporate the knowledge extracted from the environment in its knowledge bases as percepts, contexts and schemata; and

b) in the sense that could be applied different platforms almost without modification and given the "proper" sensors and actuators for the goal assigned it will learn schemata about its own operation and its interaction with the environment.

Astro-baby discovers Bang-Bang Control and applies it efficiently to perform the assigned task.

GLOSSARY

An attempt is made to formally and concisely define terminology used frequently throughout this paper, so as to minimize interdisciplinary misunderstandings.

Resolution - The granularity at which a particular situation is viewed based upon the size of a minimum distinguishable unit of space

Multiresolutional System - A system which views the world at multiple levels of granularity

Multiresolutional Hierarchy - A graph-like structure used to demonstrate the organization of data in a multiresolutional system

Learning - The process of acquiring knowledge about the world and developing behavior patterns to deal with accomplishing a specified task within the framework of the acquired knowledge

Bootstrap Knowledge - An initial set of information or knowledge, including, more specifically, techniques required for learning

Goal - A desired outcome of events

Intelligence - The ability to efficiently process and organize knowledge acquired through learning

Intelligent System - A system exhibiting the properties of intelligence and using them for control

Task Decomposition - A process whereby a given Goal is subdivided into sub-Goals which are achievable at a particular level of Resolution in a given temporal discrete. Often used by Intelligent Systems to reduce computational complexity

Percept - A set of sensor values acquired at a particular level of resolution at a particular moment of time

Context - Various information about the world at a particular time. Context may include Percept information, as well as data from other sources

Action - A set of activation of actuators in a body, to perform a task, usually set by a Goal

Situation - A grouping of information about the world and the task at hand, consisting of a Percept, a Context, and a Goal

Schema (pl. schemata) - A logical operation relating a Percept, Context, and Goal with an Action. Can be expressed as follows: IF (Percept & Context & Goal) THEN Action

REFERENCES

- [1] Warren, Charles W., "A technique for autonomous underwater vehicle route planning", IEEE J. Oceanic Eng., v.15 n.3, July 1990 pp. 199-204
- [2] Meystel, A., "Baby-Robot: on the analysis of cognitive controllers for robotics", Proc. IEEE Int'l Conf. on Systems, Man, Cybernetics, Tucson Arizona, 1985
- [3] Eilbert, J., Meystel, A., Venetsky, L., Zietz, S., "Baby Robot: Learning to control goal-oriented behavior", Proc. IEEE Workshop on Intelligent Control, 1985
- [4] Mataric, M.J., "Integration of representation into goal-driven behavior-based robots", IEEE Trans. on Robotics and Automation v.8 n.3, pp. 304-312
- [5] Maximov, Y., Meystel, A., "Optimum Design of Multiresolutional Hierarchical Control Systems," Proceedings of the 1992 IEEE International Symposium on Intelligent Control, pp. 514-520, Glasgow, Scotland, U.K.
- [6] Drescher, G.L., *Made-Up Minds*, MIT Press, 1991
- [7] Arbib, M.A. "Modelling neural mechanisms of visuomotor coordination in frog and toad", *Competition and Cooperation in Neural Nets*, Volume 45 of Lecture Notes in Biomathematics, Chapter 21, Proceedings, Kyoto, Springer-Verlag 1982
- [8] Lyons, D., "RS: A formal model of distributed computation for sensory-based robot control", COINS Tech. Report 86-43, UMass Amherst, 1986
- [9] Arbib, M.A., *The metaphorical brain 2: Neural networks and beyond*, "Schema Theory" (Chapter 5), John Wiley, 1989
- [10] Meystel, A., "Multiresolutional schemata for motion control", Drexel Univ. IMPAQT Center Technical Report, 1993
- [11] Meystel, A., "Knowledge-based nested hierarchical control", *Advances in Automation and Robotics*, pp. 63-151, JAI Press 1990
- [12] Floyd, C.A., Kanayama, Y., Magrino, C., "Underwater obstacle recognition using a low-resolution sonar", Seventh Int'l Symp. on Unmanned Untethered Submersible Technology, September 1991