S27-60
208
188109

Nested Hierarchical Controller With Partial Autonomy

A. Meystel
Drexel University
Philadelphia, PA 19104

D4178438

Abstract. The problem of computer architecture for intelligent robots with partial autonomy is addressed. Robot with partial autonomy is considered a degenerated case of a fully autonomous robot. Thus, the problem of man-machine communication is formulated, and the conditions are determined for generating a language for such a communication. The duties of the master are determined.

Key words: *Intelligent Control, Robotics, Man-machine Operation, Hierarchical Controller, Autonomous Robots, Nesting.*

1. Introduction

Problem of partial autonomy. The problem of partial autonomy is usually considered to be a problem of enhancing the capabilities of a teleoperated pobot. Partial autonomy is meant to compensate for the deficiencies of human perception and/or performance (DHPP) when the CRT is used as a "narrow window" to the world, and no or limited other means of perception are available. After the partial autonomy emerges as a necessary property of an intelligent robot, one can expect to utilize it also for simple repetitive jobs, other routine operations, in other words, as a tool of **automated** operation rather then **autonomous** operation.

The first term (automated) means independent operation of a machine, following a set of prescribed decisions which are submitted by a human operator in advance. In other words, automation presumes existence of an explicit, and/or implicit program of operation: all decisions are already made, and all possible situations are taken in account. The second term (autonomous) presumes that there are stages of planning and decision-making performed by the machine with no human involvement. In fact, autonomy means that the set of operations to be performed (or "jobs") is not well structured, more than one possibility exist, and it is up to the robot to select which one is more beneficial.(This can happen not only when the field of view is narrow and/or cluttered by unknown objects. For example, making a step upon an unstructured surface by a multilink redundant leg, generates a huge set of ill-defined problems. Human operator cannot and need not share his attention among all these problems).

The problem of joint design of man-machine control system was first addressed in [1]. Based upon the notion of a limited "bandpass required of a man" the paper suggests that the control system should be allowed autonomy in "integration, differentiation, feedforward", and a number of other computation procedures in order to allow a man "to operate as a simple amplifier". The fact that a human operator still has to perform a unique job of information integration and decision making, at this stage was overlooked. However, very soon the human capabilities to be a universal feedback with unlimited "sensor fusion" talents were questioned in [2].

Rapid growth of the requirements to productivity and quality factors, brought DHPP mentioned above to the attention of researchers in the area of advanced control systems. The variety of teleoperated and supervised automated systems described in [3-5] concentrate upon enhancement of human capabilities by allocating some properties of autonomy within the machine control system. The state-of-the-art reflected in the literature, implies the sequence of development stages as shown in Figure 1. A need in the Interactive Manual-Automatic Control arises in a natural way due to the DHPP. According to [5], manual-automatic control allows for some motions under the manual control whereas the remaining motions are performed automatically referenced to a variety of sensor data. One of the systems created at JPL, features a menu which can be activated by an operator in modifications starting with fully manual and ending with fully automated operation.

However, the system with partial autonomy can be approached in a different way. Let us imagine that the structure of a system is available with full autonomy of operation. How does this structure look like? What are the components of this structure that can and cannot be achieved by the technological means under consideration? It may happen that some elements of the autonomous operation can never be achieved, or can be achieved in a very remote future. The set of such "unachievable-now" operations can be considered **domain of impossibility of the mission** (DIM-area).Then it would be reasonable now to make a step back from the ideal image of the autonomous machine, and build a system which rely on human participation however only within the limited part: within the DIM-area.

This approach to the problem of partial autonomy can be illustrated by Figure 2. We can see that the supervised automation is viewed as a retreat from the demand for a fully automated system. In the same vein, the partially autonomous, or supervised autonomous robot can be considered a trade-off between the demand for a fully automated system (which has never been satisfied), and a demand for a fully autonomous system (which cannot be saisfied, at least currently).

PAGE 250 INTENTIONALLY BLANK

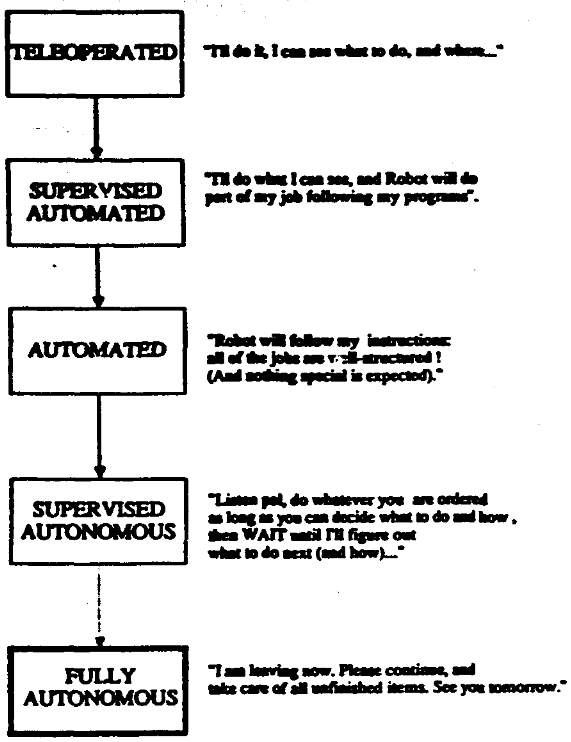

Figure 1. Typical approach to to the problem of partial autonomy

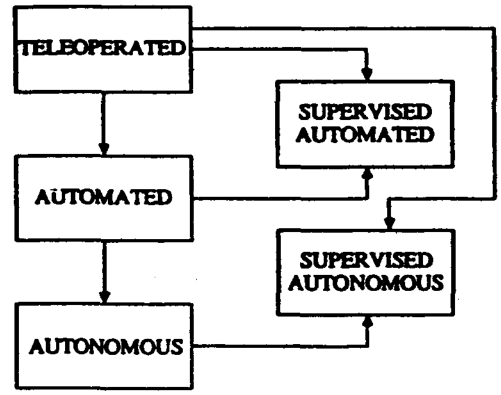

Figure 2. Our approach to to the problem of partial autonomy

Focus of concentration Problem of autonomous robot operation is often erroneously considered a problem of intelligent perception. In fact, even having the problems of perception and knowledge organization solved, the problem of motion planning and control of autonomous as well as partially autonomous robots remains unresolved. This paper attempts to formulate methods of theoretical analysis for partially autonomous planning/control processes of a class of intelligent robots equipped by a control architecture designed for fully autonomous operation. This control architecture has an important distinction which affects supervised operation with partial autonomy. Areas of motion planning, and motion control are treated separately in non-autonomous systems where such a practice is acceptable . Autonomous systems are intrinsically based upon a unified computer system jointly emulating planning-control activities as a part of a unified recursive computational process.Theory of joint planning-control systems and processes, applicable for robots equipped with Autonomous Control Systems (ACS) ascends from the theory of decision making in the framework of the control systems theory.

ACS were introduced first in [6-9], and the elements of ACS theory are presented there. ACS are to be installed in autonomous robots which should operate in unknown environment with limited human involvement or with no human involvement at all. ACS are expected to function using human-like procedures of perception. They have to maintain sophisticated information structure capable of dealing with learned knowledge, and communicating with so called, knowledge based control systems (KBCS). We make a distinction between knowledge, information, and data. Knowledge is understood as a structured information incorporating numerical data as well as linguistical, or symbolic information, which must be interpreted in a definite context. It will be demonstrated that the principle of nested hierarchies allows for an efficient knowledge organization in KBCS as well as for correspondingly efficient processes of knowledge-based perception and control.

II.Joint Planning/Control Process

Control solutions to be considered The word *control* is being used here in its initial meaning for decision making activities including *planning*, *navigation*, and *guidance*. This triad is presumed to be applicable at least, to systems for control of mechanical motion. It should provide a mapping of the a)task,b)knowledge of the system, and c) knowledge of the environment - into the output specifications.This triad is an inseparable whole, and each of the elementary parts of this triad can operate properly only in cooperation and coordination with the others.

An overview of the recent results in the area of Autonomous Control Systems for mobile robots [10], suggests that different control solutions show definite resemblance with human teams: they have a hierarchy of decision-making units even when the system is equipped by a single actuator [11]. In Figure 3, three examples of control structures are compared [12-16]. Each of these structures consists of a number of goal seeking decision units, each tends to form an intelligent module for automatic generation of control strategies, policies, and commands.

However, the most remarkable thing about all of these systems is that in all of them the scope of decision making is becoming smaller, and the resolution of decision making is becoming higher when the level of decision making is becoming lower. This sequential multiple process of decision making (replanning, and finding a new control sequence) in the same situation at different resolutions, and with different scope of attention, seems to be almost obvious. The major theoretical and design recommendations are hidden in this fact, and thus usually escape from our attention. We would assume that the underlying theory should be applicable to ACS.

The major result pronounced within the theory of team control as applicable to robotics, is related to the information structure required to run the system. The realization of this information structure is a

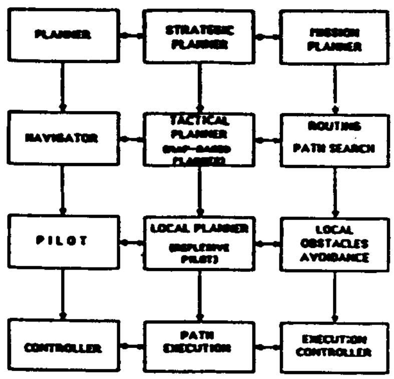

Figure 3. Comparison of control structures proposed for Autonomous Robots by the authors of [8-12].

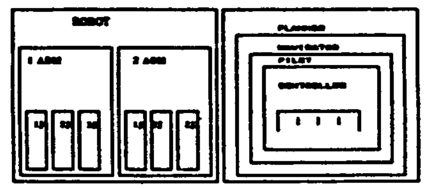

Figure 4. On comparison between nested tree-hierarchy (a), and nested stem-hierarchy (b)

special, control oriented knowledge structure. (This subject was addressed in [16], and familiarity with the treatment of the problem given there is presumed). Theory of control oriented knowledge organization as a part of the ACS, is focused upon development of models of KBCS for motion, structures of algorithms, and design of systems for optimum motion of autonomous or semiautonomous systems. Theory of ACS employs the fact that the similarities among the existing structures of control (mostly, knowledge-based) for autonomous robots reveal a number of inner mechanisms of goal oriented dealing with knowledge, as a part of ACS functioning.

Roughly speaking, any ACS is organized as a ***team of human decision-makers*** which allow for using many efficient solutions developed for human teams. Using the theory of team-control an efficient procedure of computation can be arranged which enables real-time operation of ACS. Thus, emphasis is done on solutions which are tractable, and do not lead to NP-hard problems. In the knowledge based controllers this problem gets a specific content: information structure should be suited a proper knowledge quantization. Some steps in this direction were made in [6-9].

Hierarchical structures of attention and resolution. In this subsection we will consider some of the peculiar properties of hierarchical control systems. Hierarchical decomposition of systems is a well known phenomenon, and is used usually as a method of dealing with their complexities. A system is being decomposed in parts (partitioned) when the parts contain some active elements of the systems subject to control (actuators). Thus, it is typical to consider multiactuator control systems in a form of "tree-hierarchies". Most of the statements in the theory of intelligent control, are formulated for the "tree-hierarchies" of the multiactuator systems, where the hierarchical decentralized techniques are recommended: relative independence of the levels is as important, as coordination of the branches [17-21]. The hierarchy is being preserved by the fact of resolutional as well as attentional ***nesting***. A typical decentralized control system allows for a tree-decomposition which leads to a nested hierarchy exemplified in Figure 4,a.

A degenerated case is turned out to be of substantial importance: when no multiplicity of actuators exist, and yet, a single actuator needs a nested stem-hierarchy of decision-makers in order to be properly controlled (e.g. shown in Figure 4,b). Usually, hierarchical systems are convenient as a form of organization for large systems which contain a number of goal-seeking decision units (subsystems). Thus, the problem usually includes coordinating their actions as to optimize the process of goal achievement.

Decision making processes in a nested hierarchical structure. Decision-making process which is being done upon the nested set of representations can be characterized by different frequency, accuracy, and combinational power. However, this difference can be stated only if the absolute values of these criteria are compared. Indeed, the frequency of decision-making is the lowest at the representation "a" (2.10^{-4} to 10^{-3} Hz), is higher in representations "b" and "c" (10^{-3} to 10^{-1} Hz), and is the highest in representation "d" (10^{-1} to 5Hz).

Certainly, if the zooming is being continued, the subsequent "magnifying the image" will bring attention to the processes of controller compensation, and the frequency of decision-making will grow even more. On the other hand, the accuracy of decision-making is the lowest for the representation "a": the error of decision can be in miles, and then decreases down to inches in figure "c". Certainly, on the level of controller compensation, the required accuracy might be even higher. Speaking about "combinatorial power", the solutions of representation "a" can lead to substantial differences in the overall operation, while the variety of alternatives for the narrow scope of representation "c" does not seem to be as drastic as in the case of "a".

However, all these matters assume a totally different consideration. The real difference between representations "a" through "c" is in the resolution of representation. Indeed, resolution (or the smallest distinguishable element of the image) in the case "c" is small, and in the case "a" is large, practically, the values of resolution are determined by the values of the accuracy allowed). First, we notice that the number of "objects" of the world represented in all of the images is the same due to the difference of resolution. Secondly, we can easily find that at a given frequency of decision-making at a level of consideration, the moving robot (speed is constant) will pass during the period of time between two consecutive decision-making processes, the same quantity of resolution units so that the ratio

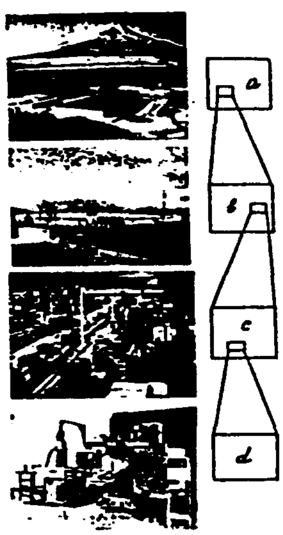

Figure 5. Nested Hierarchical Perception of the assignment for a robot

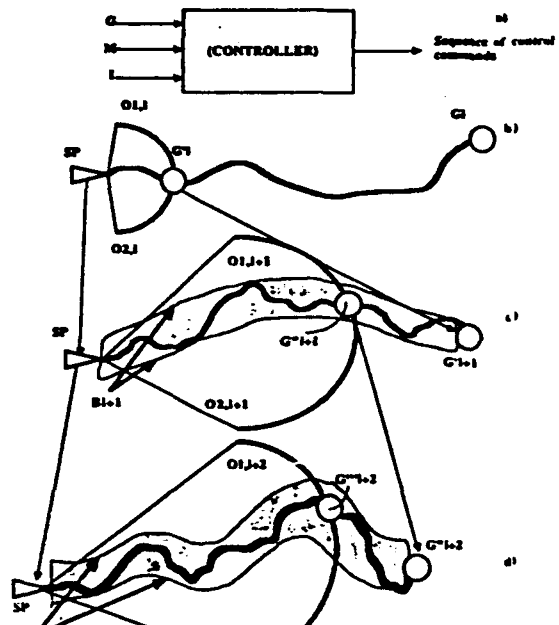

Figure 6. Joint Planning/Control Process

$$\frac{\text{(maximum state space changes per decision)}}{\text{(time interval between two consecutive decisions)}} = \text{const}$$

is constant at all levels. After simple transformation we receive that

$$\frac{\text{(time interval between 2 consecutive decisions)}}{\text{(resolution at the level)}} = \text{const}$$

for all levels of consideration.

Similar consideration brings us to conclusion that "combinatorial power" is also the same at all levels of consideration; it is just applied to different units of information, units that have different resolution. This hierarchy of representations at different resolution can be easily transformed into a hierarchy of

Decision making processes of planning-control procedures In the most general form, the controller can be represented as a box with three inputs, and only one output. These inputs can be specified as follows (see Figure 6,a):

- -Task: the goal (G) to be achieved, (and conditions to be satisfied including the parametrical constraints, the form of the cost function, its value, or its behavior).
- -Description of the "exosystem", or map of the world including numerous items of information to be taken in account during the process of control; map of the world (M) is often incomplete, sometimes, it is deceptive.
- -Current information (I) is the information set delivered by the sensors in thevery beginning of the process of control, and continuing to be delivered during the process of ACS operation.

The processes within the controller are illustrated in Figure 6,b,c,d.Input T has determined where the points SP_i and G are situated within the input map M. Input I gives a limited zone $O_{1,i}$-$O_{2,i}$(in the vicinity of SP_i) in which the information set is more reliable, or even exhaustive.Thus, in the overall planned trajectory SP_i-G'_i-G_i, a part SP_i-G'_i can be determined with more reliability than the rest of it. In other words, the overall trajectory SP_i-G'_i-G_i from SP to G might be changed in the future if the input I will update the map M in the way that the computed (planned) trajectory will become not the most desirable alternative. However, SP_i-G'_i part of the plan (bold line) won't be changed: no new information is expected.

Let us concentrate now on the part of the trajectory SP_i-G'_i which is assumed to be more known than the rest of the overall plan.Label the point G'_i the "new goal", or the "goal for the lower level of the nested hierarchy". Notice that instead of the planned curve segment SP_i-G'_i the stripe is known with boundaries $B_{.+1}$.Within this stripe, a new planned motion trajectory can be determined at a higher level of resolution pertained to the level (i+1). Again, only part of the trajectory SP-G"i+1 can be refined within the sector $O_{1,i+1}$-$O_{2,i+1}$. This in turn, brings us to the more precise part SP_{i+1}-$G"_{i+1}$ which can be subsequently refined at the level (i+2). Enentually "new goals" of the adjacent levels are becoming indistinguishable and the the trajectory SP-Gn can be utilized for computing the actual control sequence. All previously determined trajectories are plans at different level of refinement.

This simple consideration includes the following components of the joint planning/control process .

1.Finding the optimum plan SP-G based upon the map M, ant the task formulation (SP, G, cost-function, constraints).
   - 1.1 Computing the alternatives of SP-G.
   - 1.2 Comparison of the available alternatives.
   - 1.3 Selection of the preferable alternative.

2.Updating the map information M in the vicinity of SP by using the sensor information I.
   - 2.1 Recognition of the set I.
   - 2.2 Comparison between M and I.
   - 2.3 Deciding upon required changes in the cases of: $M/I\neq\emptyset$, $I/M\neq\emptyset$, $I=M$.
   - 2.4 Creation of the new map in the vicinity of SP with required deletions and additions.

3.Refined planning the path within the updated zone of the map.
   - 3.1 Determining the subgoal G' (e.g. as a point of intersection of SP-G and the boundaries of updated part of the map).
   - 3.2 Finding the optimum plan SP-G' (which implies that the whole set of procedures mentioned in p.1, should be repeated).

4\. Track the optimum path SP-G'.

5\. Upon arrival to the new point of the selected trajectory (distinguishable from the initial point) loop to the position 2.

The set of procedures 1 recursive through 3 is illustrated in Figure 6,b. Easy to notice, two major recursions are presumed within the structure of accepted decision making activities. Firstly, position 3.2 generates recursion which presumes consecutive refinement of the information within the zone which allows for this refinement. Secondly, the major loop from 5 to the position 2 presumes repetition of the whole set of the procedures after each increment of the motion. Thus, the process of path planning can be substituted by a consecutive determining of subgoals located closer and closer to the current position.However, still remains unclear a) how to determine the trajectory SP-G, and b) how to determine the zone in which the refined information should be obtained. These two key elements of the overall process will determine the number of recursions to be performed for the each point of the motion trajectory.

Substitution of positioning by tracking Theory of tracking control is much better developed than the theory of positioning (or control of "pick-and-place" operations). No wonder, we are interested in reduction of positioning problem to the tracking problem. The similarity between our control and "tracking the target", is becoming even more noticeable after we realize that the tracking trajectory is being constantly recomputed in the process of the above mentioned recursion.

Obviously, our best planned trajectory can be considered as a predicted trajectory. At the each level of the system shown in Figure 6, the bold part of the trajectory is a plan per se, and the rest is just a predicted trajectory. However, for the next consecutive lower level the situation is becoming different. The plan of the upper level is becoming just a planning envelope for the lower level. Within the **planning envelope** assigned by the upper level, **an optimum motion trajectory** is being refined which in turn is considered as a plan for the adjacent level below. The initial part of this trajectory is considered the low resolution control to follow (1-st part of the plan) and the rest is becoming just a prediction (2-nd part of the plan).

Since part of SP-G is cosidered as a prediction anyway, and since the corresponding information is to be updated in the future, the problem of system's readiness to incorporate the results of updating information, can be discussed. This generates such topics as storing the alternatives of plan, evaluation of predictions by some probability measure, and synthesis of contingency plans.

Commutative diagram for a nested hierarchical controller. Hierarchical decision making process allows for using efficiently the full computer capacity which is limited at each level of such hierarchy (with no branching). In this case, the tree hierarchy of intelligent control converts into a **nested hierarchical controller** (NHC). If NHC is acting under the above mentioned constraints the process of control allows for decoupling in parts dealing separately with information of different degree of resolution (easily interpreted as certainty, belief, etc.) This means that the degree of "fuzziness" is different at different levels of the hierarchy, and in the nested hierarchy of the fuzzy-state automata **each automaton of the lower level (automaton-child) is enclosed in the corresponding parent-automaton**, and serves for clarification of its uncertainties.

Considering **subsystems as categories C**, and the interaction among them as **functors F**, the commutative diagram of the ACS can be shown in Figure 7 (indices mean: s-sensing, p-perception, k-knowledge, pc-planning/control, a-actuation, w-world). Feedbacks are not shown: boxes are connected by "functors" which characterize the structure conservation in a set of mappings of interest. The bold horizontal line separates two major different parts of the system: what is below, is a world of real objects, and what is above the bold line, is the world of information processing. All of the "boxes" in Figure 7 are fuzzy-state automata. They are easily and adequately described in terms of the automata theory, and provide consistency of the descriptions, computer representations, and control operation. Then, the search can be done by combining A* and dynamic programming, discretization of the space is being determined by the level of resolution, and the rules which are formulated within the given context.

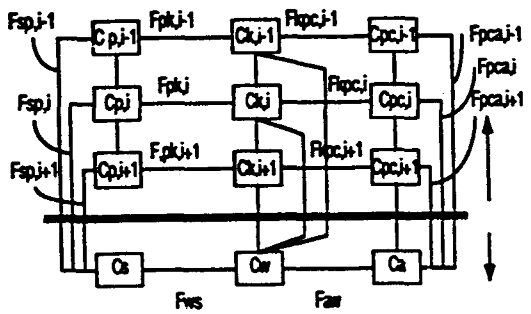

Figure 7. Category-theoretical Diagram of Nested Hierarchical Controller

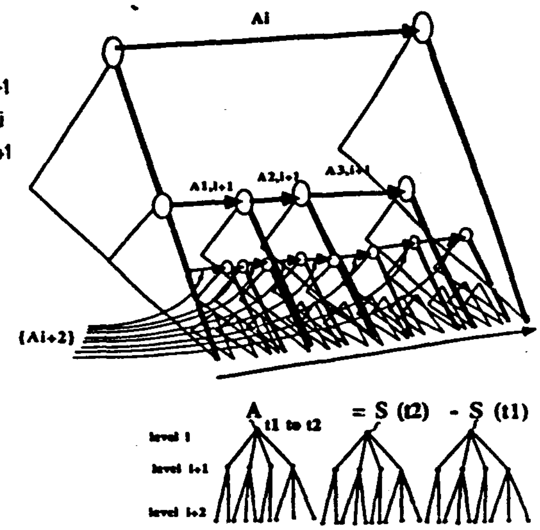

Figure 8. On the relations between representation and control (S-states, A-actions)

Nested dynamic programming The principle of optimality of Bellman [22] can be stated as follows for stochastic problems: at any time whatever the present information and past decisions, the remaining decisions must constitute an optimal policy with regard to the current information set [23].Thesequence of "planning-navigating-piloting (or guidance)-actuator control (or execution)" appears as a direct result of nested hierarchical search in the structure of information under consideration. Method of Nested Dynamic Programming (NDP) follows from the commutative diagrams and analysis given in [6-9]. It states that the optimum control should be found by consecutive top-down and bottom-up procedures, based on the following rules.

Rule 1. NDP should be performed first at the most generalized level of information system with complete (available) world representation.

Rule 2. NDP is being performed consecutively level after level top down. The subspace of the search at each of the consecutive lower levels is constrained by the solution at the preceding upper level recomputed to the resolution of the next lower level.

Rule 3. When during the actual motion, due to the new information, the optimum trajectory determined at a given level must violate the assigned boundaries, this new information should be submitted to the upper level (proper generalization must be performed, and the information structure must be updated). This generates a new top-down NDP process.

Rule 4. When arrival of the new information is bounded (e.g. by a "limit of vision"), then the recursion of nested process of planning is being done with consecutive process of subgoals creation.

Control Stratified by Resolution into a Joint Multilelel Planning/Control Process. The nested hierarchy of maps $\{m_i\}$,i=1,2,... is the input for planning/control system. Actually, this nested hierarchy is being generated in the process of interaction between the subsystems M and C . Let us show how this process of interaction is going fa g fa g g fa g fa

line1: $(m_1 \in M_1) \supset (m_2 \in M_2) \supset \dots \supset (m_{i-1} \in M_{i-1}) \supset (m_i \in M_i)$

fa fa fa fa

$\downarrow$ ... ... $\downarrow$ $\downarrow$

line2: $NDP(J_1, M_1) = m_1$ ... ... $NDP(J_{i-1}, M_{i-1}) = m_{i-1}$ $NDP(J_i, M_i) = m_i$

$\downarrow$ ... ... $\downarrow$ $\downarrow$

line 3: **output** ... $CH(m_{i-1}) = M_{i-2}$ $CH(m_i) = M_{i-1}$

$\downarrow$ $\downarrow$ $\downarrow$

to controller ... **to (i-2)-th** **to (i-1)-th**

l e v e l s

Line 1 shows two nested hierarchies: one of them by generalization (of maps $\{M_i\}$) and another by focus of attention (of maps $\{m_i\}$). Hierarchy of sets is obtained from the hierarchy of sets by applying NDP-algorithm per level (line 2). In order to do this, a hested hierarchy is added to the nested hierarchy . In order to compute a set ,the results of applying NDP per level, are enhanced up to the meaningful consistent map partition; one of possible algorithms for this is "convex hull" (CH). So this system is closed loop level to level top-down and after convergence, the system of controller commands is obtained.

This planning/control algorithm represents actually a **hierarchy of blackboards** (HBB) in which the communication is being rendered among interacting subsystems of the intelligent module.**Main navigator** takes this plan as a suggestion of the milestones, and determines a string of subgoals at the way to the first of milestones. This string is sent to the **pilot** which also has nested structure.The first subgoal in this string might be invisible for the ACS, the visible goal is to be determined by the **planner within the pilot**. Then **pilot's navigator** can compute accurate enough trajectory of motion to be executed. This trajectory is submitted to **controller** which in turn, has nested structure.Its **planner** determines the next tarcking command, then its **navigator** determines the execution sequence, etc.

Built-in combinatorial operators:team of decision-makers Not only strings or combinations of variables (words) as well as combinations of mappings (clauses) will be considered, but also combinations of these combinations. Search by scanning, and inclusion when the desirable property is met, is one of the straightforward algorithms of combination generation. If the results of search are constantly enhancing the input vocabulary, then during the exhaustive search with recursive enhancement of the vocabularies, all possible unions, intersections, and complementations of the sets are being obtained. The final vocabulary forms a field (F) which means that it contains all set of subsets constructed upon the initial set including the results of applying all acceptable combinatorial operations, (one of these operations should be later used for decision making process: combinatorial search). Using focus of attention we receive a special category of **limited fields** which do not contain all combination of components which is required by a formal definition of field, and yet gives more meaningful recommendations without introduction of complicated algebraic structures.

Using search for alternatives generation. Any combinatorial algorithm is an operator of generating plans or **solution alternatives** for a decision-making process. Consider A*-algorithm for the search of minimum path trajectory [29,32,35], or any type of conventional or 'enhanced" dynamic programming [22,23,31,33]. Then a number (value) is assigned to each of the combinations generated (preferability, closeness, propensity, cost-effectiveness, etc.) which will enable the decision-maker to make his choice under the accepted **strategy of decision making**. According to the existing terminology, the chain of consecutive decisions named **policy** of decision-making, or policy of control

Since we have received already a hierarchical structure of representation, and N levels are considered which are representing the same world with different resolution, the following situation should be considered. Given a vocabulary consisting of N elements at the level "i" (words) $w_{1i}, w_{2i}, \dots, w_{Ni}$ the set S_i of admissible decisions (or decision N-tuples) determined within the boundaries determined by the upper level $B\{w_{i-1}\}$

$$S_i = w_{1i} \otimes w_{2i} \otimes \dots \otimes w_{Ni} = \{w_i\},$$

and assume the cost-functionals (e.g. presented as distances)

$$J_i : \{w_i\}, B\{w_{i-1}\} \to \rho_i$$

then the obtained N-tuple (a string $w_1 w_2 \dots w_n$) is the i-level hierarchical solution. This string can be interpreted as the next subsequent state of the world, or the change leading to the next subsequent state of the world (action). In the latter case, the subactions at various levels of the hierarchy, show the overall action which leads from one substate to another. Clearly, any decision making process is a result of decision making activities of a couple of adjacent levels, which means that the planning/control procedure in a result of a recursive computation upon the whole hierarchy of representation. Strictly speaking, the decision making at a level require involvement of at least two adjacent levels. Thus, the structure of the levels representation is becoming of importance.

II. Representation System

Information set contained in the representation system. The information set at time k is assumed to be the past measurements and controls for all levels of our hierarchy nested by

a)generalization:

$$(I_i(k_i) \overset{g}{\supset} I_{i+1}(k_{i+1}) \overset{g}{\supset} I_{i+2}(k_{i+2}) \overset{g}{\supset} \dots) \therefore$$

$$\{Y_i(k_i) \overset{g}{\supset} Y_{i+1}(k_{i+1}) \overset{g}{\supset} \dots, U_i(k_{i-1}) \overset{g}{\supset}$$

$$U_{i+1}(\overset{g}{k}_{i+1}\overset{t}{-1}) \supset \dots \overset{g}{\}} \supset I_i(k_{i-1}) \overset{g}{\supset} I_{i+1}(k_{i+1}-1) \supset \dots$$

b)focus of attention

$$(I_i(k_i) \overset{fa}{\supset} I_{i-1}(k_{i-1}) \overset{fa}{\supset} I_{i-2}(k_{i-2}) \overset{fa}{\supset} \dots) \therefore$$

$$\{Y_i(k_i) \overset{fa}{\supset} Y_{i-1}(k_{i-1}) \overset{fa}{\supset} \dots, U_i(k_i-1) \overset{fa}{\supset}$$

$$U_{i-1}(k_{i-1}-1) \overset{fa}{\supset} \dots \} \overset{t}{\supset}$$

$$I_i(k_i-1) \overset{fa}{\supset} I_{i-1}(k_{i-1}-1) \overset{fa}{\supset} \dots$$

It is shown in [5-9] that the phenomena of nesting by generalization and focusing the attention can be characterized best by the operators of inclusion which presumes some procedures unusual for the conventional approaches to control systems. Indeed, nesting by generalization is obtained as a result of a)grouping the object of the i-th level by the class-generating feature, b)assigning a label to this group, c)placing this laber into the (i-1)-th level as a single object, d)repeating the same for the words of the (i-1)-th level,and e)finally, obtaining a new word at the (i-2)-th level.Then, any class of the (i-1)-th level is **nested by generalization** in its much broader representation at the i-th level.On the other hand, after a **solution** is obtained as a string within the class of the (i-1)-th level, this solution will automatically **"carve out"** a subset within the corresponding much broader representation at the i-th level, of the same (i-1)-th level class in which the solution was determined.The result of such "carving out" is **nested by attention** within its own class of the i-th level.

Partitioning and nesting of representations. Decomposition of the categories of representation is done through decomposition of objects and morphisms represented in this category [24]. Decomposition imlies dividing objects and relationships in parts, so the components of objects are becoming of interest which was not the fact before the process of decomposition. This, in turn, implies higher resolution of world representation than before the process of decomposition. If we continue with decomposition, the further representations are expected to disclose new details about the objects and their parts, as well as about relations and their parts. So, in the hierarchy of decompositions, all of the levels describe the same world with higher and higher level of resolution.

Changes in time are represented by sequences of the snapshots in domain of representation describing sequences of the world states. Thus, change in the time-scale is intrinsically linked with the value of resolution. Certainly, different levels of the hierarchy can be characterized by different time-scales. Allthese phenomena are illustrated in Figure 8.

Nested Tesselation of Knowledge. This system of nested variables and nested mappings is described in a Hausdorff nested H-space which means that any two different n-dimensional points in it have at each level of nesting, disjoint neighborhoods. One may assume that our space is obtained as a result of natural, and or artificial discretization (tesselation, partitioning) from an initial continuous space of the isomorphical world description. This space can be considered as a nested state-space for a dynamical system in which the trajectory of motion can be represented. This space is considered for a given moment of time (snapshot of the world). Then, a sequence of snapshots can be obtained. The conventional state-space is a superposition of all snapshots which describe the process.

Two types of discretization (natural, and artificial) should be understood as follows. After the problem is formulated, it is already based upon some world discretization since the context is represented by a semantic network discretizing the world in a "natural" way. However, when the higher level of resolution is desired, the natural discretes can appear to be insufficient. Then another "artificial" type of discretization is applied, and the space is tesselated by (say) a "mechanical", quantitative division of larger bodies into the smaller parts.

This tesselation is done in such a manner that the "points of interest" (symbolizing the variables, words, or wwf's) are placed to the center of the "tile" of tesselation at a level (elementary segment, grain, discrete, pixel, or voxel of the space). Such terms as segmentation, granularity, discretization, and tesselation can be used intermittently. Property of being centered, holds if the tile can be derived as a system of subsets with unempty intersection. This property should be considered rather in a symbolic way: in fact, we cannot discern any another point within the tile, this is a minimum grain at this level of resolution.

Resolution of Knowledge In other words, the tile of the tesselation determines the **resolution of knowledge** which is defined as a minimum discrete of information, or minimum wwf which can be stated unmistakably.The minimum centered tile will have diameter ε and the net of centers emerging from this tesselation is named ε-net [25].Analytical details can be found in [6-9].

The idea of nested tesselations is coming together with a definition of a single tile $T(\varepsilon)$ based upon nested sphere theorem which can be rephrased as nested tile theorem which defines a chain of inclusionsfor the hierarchy of tesselations. Thus, ε -net is considered to be a set of elementary tiles satisfying a condition

$$T(x_0, \varepsilon_0) \supset T(x_1, \varepsilon_1) \supset T(x_2, \varepsilon_2) \supset ... \supset T(x_n, \varepsilon_n)$$

where $x_1, x_2, ..., x_n$ are the coordinates of the centers of the tiles,

$\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$ are the radii of the nested tiles.

In the equation of relationships among the tiles

$$\varepsilon_0 = \frac{\varepsilon_1}{\sigma_1} = \frac{\varepsilon_2}{\sigma_2} = ... = \frac{\varepsilon_n}{\sigma_n}$$

Pairs of adjacent levels. Hierarchies created in this way, satisfy the following principle: **at a given level, the results of generalization (classes) serve as primitives for the above level.** Then each level of the hierarchy has its own classes and primitives; thus, it has its own variables (vocabulary), and the algorithms assigned upon this vocabulary can be adjusted to the properties of the objects represented by the vocabulary. This determines the mandatory rule of combined consideration of two (at least) adjacent levels of the hierarchy. Set of relationships among the variables at the i-th level describes the implications which are being utilized for decision-making at this particular level.

On the other hand, the set of relationships at the (i+1)-th level describes the relationships among classes and thus can be characterized as set of **metaimplications** (metarules, metaclauses) which are governing the process of applying the implications (rules) at the i-th level. In the light of this consideration, each two adjacent levels can be understood as a complete E. Post production system (analog to "general problem solver" or "knowledge based controller") in which the metarules applied to the alphabet of the lower level act as a semi-Thue process of the grammar [26].

"natural" (qualitative) way (based upon "words" of the context), and the lower level is discretized in an "artificial" (quantitative) way (based upon grids, etc.). The process of decision making in this case depend on the ability of these two levels to communicate efficiently despite of the different representation of the elementary "tiles". The role of human "master" or "supervisor" is critical for operation of such adjacent levels since he is the real carrier of the context knowledge which is required for the adequate translation from the language of the (i-1)-th level into the language of i-th level and vice versa.

Resolution of representation. Labeling the class presumes dealing with this class as a primitive at the given level of consideration. Moreover, this class (now, also a primitive) is being again clustered in classes of the higher level. In order to deal with class as with a primitive we have to neglect the inner content of this class (which might be reflected as new properties of the new primitive but with no mentioning the initial primitives with their properties). The levels of the hierarchy of representation (since they are created by a mechanism of generalization) are dealing with the same world given with different level of richness in submitting specific details, level of coarseness (fineness). We will name this characteristics of world representation at the level, **resolution** which is a measure of distinguishability of the vectors in the state space.

Accuracy versus resolution. It is clear that after assigning to the cluster a new class-label (a word to be included to the vocabulary), this class-label is becoming a new primitive with its own properties, the numerical values are assigned to these new properties, and these numerical values are characterized by the accuracy depending on the accepted evaluation theory for this particular property (including probabilistic approaches, fuzzy set theory, etc.). Clearly, this accuracy evaluation is formally independent from the act of neglecting the inner contents of the new primitive. This means that accuracy and resolution are formally independent. The word "formally" means: in the view of procedure to be performed.(Consider digitized images (e.g. any "map" of the world) with different base discrete of the image. The bigger this discrete is, the less is the resolution of image. However, the accuracy of representation at this level can be the same provided the accepted value of error upon the base discrete).

Thus, accuracy presumes the error evaluation in terms of the existence of difference between the world and its description **within the accepted concept of the model**. The smaller the vocabulary Σ_i is, the more different phenomena are neglected. This neglect may entail the increase in error and may not. However, the smaller $\mathrm{Card}(\Sigma_i)$, or size of the vocabulary is, the higher is the level of generalization, and the larger is the radius of the tile in the ε-net. Thus, the following relation should hold

$$\rho = \frac{1}{\mathrm{Card}(\Sigma_i)} > \Delta$$

where ρ determines the value of allowable error (inaccuracy) and $\mathrm{Card}(\Sigma_i)$ determines the value of resolution.

Thus, the information of the world which is required for decision-making, should be explicated and prepared for the subsequent decision-making processes. Map is defined as a subset of the state-space which is to be taken in consideration during the process of decision-making. Thus, map is a part of policy delivered from the leader to the follower. Map of the upper level contains the maximum subset given at the lowest resolution. Maps for the next levels of the nested map structure (top down) are obtained using the second apparatus of nesting: *focusing of attention*. Thus, only one part of the map is being activated: which is required for the current decision-making.

The system of maps has dual nesting:

firstly, we have a subsystem with nesting by generalization

$$M_1 \overset{g}{\supset} M_2 \overset{g}{\supset} \dots \overset{g}{\supset} M_{i-1} \overset{g}{\supset} M_i$$

which we name "CARTOGRAPHER", and assign to this subsystem maintenance of the information rather than active participation in the process of decision-making.

and secondly, we have nesting by focusing attention superimposed upon the nested hierarchy by generalization , and this information extracted from the cartographer $\{m_i\}, i=1,2,\dots$ is delivered to the "C" system (Figure 7) in a form of "maps for decision making".

$$\overset{fa}{(m_1\in M_1)} \underset{fa}{\overset{g}{\supset}} \overset{fa}{(m_2\in M_2)} \underset{fa}{\overset{g}{\supset}} \dots \underset{fa}{\overset{g}{\supset}} \overset{fa}{(m_{i-1}\in M_{i-1}} \underset{fa}{\overset{g}{\supset}} \overset{fa}{(m_i\in M_i)}$$

All fa-predicates are performed on the basis of NDP-algorithm results and thus belong to the subsystem of control, all g-predicates are prerogative of the system of map maintenance. Later we will show how these predicates are built in the algorithms of planning/control and map maintenance.

The problem of map maintenance is of scientific and practical importance. The upper level map ("planner's map") should be maintained for a long time due to the largest scope, and the "slow rhythm" of this level. Changes in the upper level map are not frequent. Maps of the subsequent levels are to be regularly updated with increasing frequency but decreasing volume. At the level of "pilot's" map, most of the information might be considered of little importance in long terms, and the map is an "ephemeral" one. The lowest level map ("controller's map") may or may not need even an ephemeral map maintained as a part of the nested hierarchy. (Actually, from our first experience of dealing with ACS we found that intelligent module cannot afford maintenance of the pilot map , i.e. of the lowest level of world representation), and therefore all processes related to the real time operation have ephemeral structure indeed, with a number of logical filters determining whether this ephemeral information contains anything to be included in the maps of the upper level.

The rules of assigning for the information to be retained in details, retained in generalization, or dropped, cannot be formulated consistently for all possible situations. How long the information will be needed can probably be determined only by a **master**.

Generalization: belonging to a meaningful class. So, knowledge in a context can be represented as an ε-net at a definite resolution, and as a system of nested ε-nets with different scale where scale is defined as a ratio between resolution at a level and resolution at the lowest level. Clearly, each of the larger tiles at the upper level is a "**generalization**" for the set of smaller tiles at the lower level of the hierarchy. Selection of one of the tiles at the upper level (*focusing attention*) entails selection of a definite subset of smaller tiles at the lower level.

We would like to stress the fact that the inclusion $\mathbf{X} \supset \mathrm{X} \supset x$ shown in this hierarchy of the tile embeddings has more important and broadmeaning than just "scaling of the size". The inclusion predicate $\supset$ has a meaning of "belonging to a class" (since it is assumed that the objects are unified into a set by some class generating property, or feature). One can talk about state space, space of weighted properties, and so on, and the notion of "belonging to a class of come spatial neighborhood" is becoming closer to a meaning of "generalization" as if it is understood in the discipline of logic. Then dicretization of the state space will contribute simultaneously to a) minimum required interval of consideration, b) hierarchy of **classes of belonging to a meanigful neighborhood**.

From the theory of pattern analysis and image recognition we know that selection of the meaningful neighborhood is a difficult issue, and it cannot be addressed exhaustively unless a human operator, or master is involved in the procedure of the feature assignment.

Look-up Tables. The prespecified lists of mappings upon a vocabulary, can be organized in **look-up tables** (LUT's) for convenience. For example, a control "u" can be selected from this prespecified set if LUT contains clauses (e.g. "**if S &C then u**" where S-is a state description, C-strategy selected, or cost-functional agreed upon). Minimum look-up table can contain only one mapping.

Sets of variables to be considered simultaneously are named **vectors, or strings** . A single LUT, or a set of LUT's are named **grammar (operators, transformations)** if they can be utilized to generate output strings from input strings. Input and output vocabularies together with the grammar are named **language** (over a finate alphabet), or **automaton** (free monoid generated by the alphabet). Clearly, vectors or strings are **statements** in the above mentioned **language** (built upon words of this language). The rules of combining variables into vectors, or srings, or strings of vectors, etc. are being determined by LUTs .

Special role of attention When a part $\mathbf{X}_p$ is detached from the "whole" $\mathbf{X}$ (which may happen to be category, object, and/or morphism) and the relation of inclusion holds

$$\mathbf{X} \supset \mathbf{X}_p$$

this separation of $\mathbf{X}_p$ from $\mathbf{X}$ we will name **focusing attention** upon $\mathbf{X}_p$. Sampling is one of the common methods of focusing attention.Usually we focus the attention when the subset of attention can be considered important, or is typical for the whole set. The latter case links focusing attention with mechanism of generalization.

Special role of attention is determined by its close links with the overall goal of operation. Attention always presume sacrifice of some information in favor of the information set which is considered to be more important. Thus, attention almost always rely on a set of preferences that can be ultimately formulated only by a **master**.

Generalization oriented tesselation We would like to stress the link between the generation of entries for LUT as an approximation in the space of consideration, and the problem of **generalization oriented tesselation** (discretization of the space). At the level of discretization determined by the context, i.e. determined by the words of the semantic network, the results of tesselation are always determined by the vocabulary of the **master**.

The subsequent problems of "digitization" and "quantization", or quantitative tesselation at the lower level are being solved currently by introduction of the "shah function", and other mechanisms of sampling which has been proven to be applicable in decades of development of communication and control systems. Some difficulties, such as aliasing, and similar, lead to algorithmic aggravations at the output. In the context of this paper, discretization of the space does not allow for the problem of aliasing because there is no information between the adjacent tiles of tesselation (digitization, quantization), and the information about the tile properties (values for them) is the set of average average values over the tile. Clearly, the term "average" in this context, acquires a somewhat unusual meaning of "**class generating property**".However, in most of cases the"automatic" tesselation of the space creates discrepancies when the correspondence between the two adjacent levels is to be determined. These discrepancies might be eliminated by the interference of the **master**.

Complexity of tesselation: ε-entropy .Complexity of LUT is being evaluated by computing the ε-entropy (by A.N. Kolmogorov [25]) for a space where the corresponding ε-net has been constructed

$$H_\varepsilon(S)=\log N_\varepsilon(S)$$

where S-is a space represented by LUT,with ε-net assigned,

N_ε-number of elements (nodes) in ε-net, or tiles in the tesselation.

For the system with combinatorial search as a tool for planning, the category of consideration can be represented as a power set, then trivially

$$H_\varepsilon(S) = N_\varepsilon(S)$$

Various theories of planning are based upon the idea of space tesselation (e.g. configuratin space, etc.). A multiplicity of different strategies of space tesselation is created, and numerous techniques of evaluating the distance, and the complexity are known from the literature. It is essential to understand that all of these techniques ascend to the theory and formalisms of ε-nets. In our case, minimization of the ε-entropy allows for determining an optimum number of the levels of the hierarchy of a control system.

Accuracy of representation. To all words and all edges with the corresponding strength of the bond, the values of accuracy are assigned. Accuracy is understood as the value of the relative error of the quantitative characteristics assigned to the node or the edge of the graph. This evaluation can be based on estimation of probabilities, zones of uncertainty, and/or in any another way. The approach to the world representation in this paper does not depend on the particular way of dealing with uncertainty evaluation.The value of uncertainty is assigned to the numerical values associated with the properties of the tile of representation which is considered as a minimum unit at a level.

Evaluation of the size of a tile. This is similar to assigning a definite ε-net. However, this also suggests that the minimum value of the radius of the tile, can be time dependent. In order to avoid this predicament in the following treatment, we will try not to get involved with the particular stochastic and dynamical microstructure of the ε-tile at the level of the representation hierarchy. We will define the radius of the tile by the width of the fictitious uniform distribution which has the same value of the conditional entropy as the real distribution of the variable x. Then, using the value of conditional entropy

$$H = - \int_{-\infty}^{+\infty} p_i(x) \ln p_i(x)\, dx$$

for a Gaussian the diameter of a tile is determined

$$\Delta = \sigma\, (\pi e/2)^{.5}$$

Referring to the definition of ε-net, we would suggest that this value of Δ is the minimum size of the tile radius to be assigned if the net has the same law of distribution in all of its nodes. For systems with stationary random processes, evaluation will be sufficient with no involvement in analysis of stochastic processes anymore. Certainly, this depend on the nature of what is considered to be a random component of the variable.

Nevertheless in real situations, it is very difficult to construct a level with the same law of distribution in all of its nodes. The role of master here, is either to determine the "average" law of distribution, or to allow for a net with different sizes of tiles. Based upon condition for Δ, the problem of different laws of state-space tesselation known from the theory of motion planning (e.g. polyhedra, etc.) is virtually eliminated. The geometry of a single tile becomes irrelevant, and yet the space is fully covered by these "fuzzy" tiles.

III. Hybrid Master-Dependent Representation

Knowledge Bases: Semantic Networks with context oriented interpretation Knowledge Bases (KB) are considered to be a collection of well-formed formulas (wff's) for man-machine as well as for autonomous decision-support systems of different kinds. So, it is clear that knowledge consists of linguistical discretes (entities) which are later referred to as **units of knowledge**. When we are talking about **knowledge**, the **represented knowledge** is actually meant, (i.e. we are not interested here in discussing the nature of the reality represented within the system of world representation). Nesting is being introduced often in a semantic way , and reflects semantic discretization of resolution in representation reflecting human experience (which sometimes is irrelevant).Thus, in a nested system, a nested hierarchy of semantic networks can be formulated. This semantic nesting can be interpreted within a context only under human supervision since the subtleties of the context knowledge must be properly transformed into statements of existence or absence of relationships among the entities.

Thus, under master's supervision we receive, two types of nested hierarchies for declarative wff's: existential nesting (nested statement of existence) including nested statements of objects and relations among them, and transitive nesting (nested statement of change). Both of these nested units of knowledge are presented in implicative form (nested clauses). It seems reasonable to describe the set-theoretical on one hand, and on the other hand, vector-analytical manners of describing situations (which do not differ in essence). This another form of representation in turn, can be divided in two different kinds of representation: continuous (using differential equations) and discrete (using difference equations and/or finite state machines (see Figure 9). In both cases, the property of nesting is applied.

If a hierarchy is built using the existing semantic network, then each (i-1)-th level of the hierarchy will contain generalizations of the words of the i-th level. This means that the solutions found within the (i-1)-th level will automatically "carve out" subsets within the classes of i-th levels in which the process of combinatorial search can be repeated, and a new solution can be found with higher level of

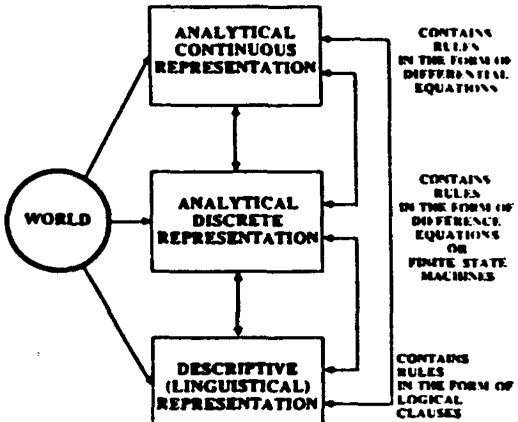

Figure 9. Different Categories for World Representation

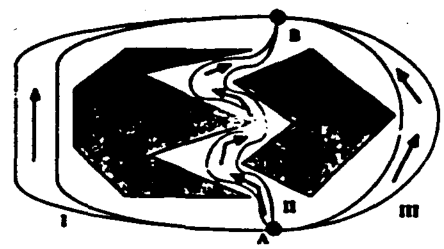

Figure 10. General solutions preceding the particular ones

resolution. However, the i-th level should **understand** the (i-1)-th message on the boundaries of the search.

Postulate of multiple representation . So, the same world can be represented within the same levels of resolution in a different way. Representation is defined as a structure (e.g. algebraic, or information structure) which is homomorphic to the world, to the structure of reality (or a domain of reality). Representation consists of both numerical as well as descriptive information about the objects and systems, and is assumed to be obtained from prior experience, and or derived theoretically (based upon multiplicity of existing and possible tools of logical inference).

The phenomenon of multiple world representation presumes that homomorphisms exist between various representations.Currently, various bodies and techniques of world representation are used in the practice of control, based upon different elements and rules of information organization. In this paper, knowledge bases as a technique of world representation, are being focused upon. Knowledge Bases as a model of information system, is a system of world representation which is considered to be **homomorphic** to other systems of world representation such as systems of difference, or differential, or integral equations, and others (see Figure 9).

Nondeterministic nested referencing. Accuracy and resolution can mistakenly be understood as nondeterministic properties of representation. We will separate the characteristics of accuracy and resolution from the nondeterministic information. Let the vector of observations be

$$X_o = X_{od} + \xi$$

where X_{od}- deterministic model after recognition,

ξ -stochastic component after observation.

Definition. A component of observation is named stochastic component if it is

a) not identified (yet) with any of models stored in the knowledge base,

b) substantially larger than measure of accuracy and resolution ($|\xi| > \rho$),

c) presumed to potentially affect the results of decision-making.

Our approach differs from the classical only in a sense of (11) which implies two recommendations.

Recommendation 1. Control problem is to be solved as a deterministic problems dealing with models X_{od} and estimating likelihood or plausibility of the decision, or the policy by measure of uncertainty linked with the set of $\{\xi_i\}$.

Recommendation 2.. Learning will be understood as a tool for extracting new recognizable models from $\{ \xi_i \}$ rather than for updating knowledge of probabilistic characteristics of the set $\{ \xi_i \}$.

The following structure of dealing with unrecognized (unmodelled) information is implied by these two recommendations. Initial decomposition should be be repeated recursively for the nested structure of information. At each level the component is being decomposed in two parts: which can be recognized and included in the deterministic part of the next level, and which at the next level remains still unrecognizable (with $E[\xi_i]=0$)

$$X_{0i}= X_{odi} + \xi_{r,i+1} + \xi_i$$

where X_{odi}- deterministic model after recognition at the level i,

$\xi_{r,i+1}$-part of the stochastic component at the i-th level which will be recognized after observation at the (i+1)-th level.("trend").

ξ_i -part of the stochastic component at the i-th level which remains unrecognized, $E[\xi_i]=0$.

This recursive analysis of the stochastic information can be illustrated as follows

$$X_{0i}= X_{odi} + \xi_{r,i+1} + \xi_i \qquad X_{0,n-1}= X_{od,n-1} + \xi_{r,n} + \xi_{n-1}$$

$$X_{0,i+1}= X_{od,i+1} + \xi_{r,i+2} + \xi_{i+1} \qquad X_{0,n}= X_{od,n}+ \xi_n$$

---

and n is the level where the recursion stops (no consequtive levels are expected to be built).

This decomposition of information (which is possible within the nested hierarchical structure) allows for multiple reference system. The key motivation for the multiple referencing is simplification of information representation per level. Multiple referencing is indirectly presented in the requirement that $E[\xi_i]=0$. This means that the origin is placed in a point in the state space as to provide $E[\xi_i]=0$. Then, the rest of the information allocated for decision-making at this level is referenced to these origin.

Another important implication of **multiple referencing** in dealing with nondeterministic information, is related to the topic of learning. As mentioned above, the system is supposed to deal with partially, or completely unknown world. Thus, learning is presumed. Any learned information is being identified with memory models (patterns) which determine the initial referencing. The residual information is supposed to be collected, and later it is expected to generate a new pattern upon the multiplicity of realizations. If generation of a new pattern seems to be impossible (no regularities are discovered), the change in the initial referencing might be undertaken. This philosophy of dealing with new information is to be utilized for procedures of **map updating**.

We can see also within the body of this problem of nested referencing, a direct link among the quantitative characteristics of the system and its linguistical description, and the components of this description. At this time, however, we will restrain from further statements on these links since we do not have enough factual observations.

Recognition and Identification. All of the above statements are based upon *Definition* containing undefined word "identified". In further considerations, the word "recognition" is used. Some list of *patterns* is presumed which should be used for comparison and identification. Recognition is also loosely defined but some freedom seems to be allowed for possible enhancement of the list of existing patterns. Most of these matters deserve a special and detailed treatment, they are not discussed in this paper. However our attitude toward this domain should be at least illustrated by an example.

An example of interpreting the situation with identification, recognition, and learning the new patterns of the entities of the world, is shown in Figure 10.If A is a robot location, and B is the goal, then minimum time path can be sought in one of the following areas: I, II, and III. It can be demonstrated geometrically, that the minimum distance path can be found in the area II. However, since each turn within this "slalom-type" area is linked with the losses of time (speed at the turning point should be reduced to avoid skidding), then the minimum time trajectory seems to be within one of the areas I or III.

It seems, that even before we start to compute all and possible trajectories within these areas, they can be analyzed as some entities with common properties. Indeed, area I can be described as an area which has "obstacle 1" on the right of the moving robot all over the motion, area IIi has "obstacle 2" on the left of the moving robot, and area II has obstacles on both sides. Areas I and III do not seem to be different when robot is moving far from the obstacle, but they differ substantially if we try to make the length of the trajectory as small as possible: the trajectory in the are I will have 5 linear segments and four turns while the trajectory III will have only two linear segments and only one turn.

Further analysis shows that the cost-functions have different form for all three areas. Moreover, within one particular area cost-function can be considered monotonoc in the sense of [32]. These kind of areas are named "topoways" [33], and they can be considered entities for minimum time problem solving at the higher level of planning/control system. Rules can be formulated in which the condition part will describe the relative obstacle location while the consequent part will be related to the most general description of the topoway recommended for further consideration.

Clearly, rules of this level of generality have to be recognized by a master, however they can be utilized with no master's direct participation.

On the equivalence between the difference equation model, and the production rules. Difference equations can be easily transformed into production rules, Indeed, instead of the form

$$x(k+1)= A(k)\, x(k) + B(k)\, u(k)$$

where k-is the number of the stage of recursion,
A(k) and B(k) are the matrices of system parameters,
x-is a state, and
u-is a control.

Instead of having this information in a form of an equation one can state

$$\text{IF } \{x(k)\wedge A(k)\wedge B(k)\wedge u(k)\} \text{ -> THEN } \{x(k+1)\}$$

which actually describes the causality exposed by a particular system : "if present state is given, and the parameters are known, and some definite control is applied, then the state will change as follows". In most of the real systems causalities obtained experimentally and/or from the other models of representation can be inverted in a form of rules-prescriptions [28]

$$\text{IF}\{x(k)\wedge x(k+1)\wedge A(k)\wedge B(k)\} \text{ -> THEN } \{u(k)\}$$

which can be interpreted as follows: "if present state is given, and the the following state is required, while the parameters of the system are known, then apply this control".

In some cases even the form of trajectory can be utilized for rules formulation. Indeed, the form

$$x(k) = F(k,0)\, x(0) + \sum_{l=0}^{k-1} F(k, l+1)\, B(l)\, u(l), \text{ and}$$

$$F(k+1,l+1) = A(k)\, F(k, l+1)$$

implies the following ways of subsequent planning/control activities:

1) solution trajectory on-line computation, submitting to the execution controller, and storing in memory for the subsequent use in a similar situation,
2) retrieving the previously stored solution for the same or similar conditions, obtained from off-line training, or from on-line experience.

Applying the production rules at different resolution levels. The above consideration can be expanded if the prior behavior is taken in consideration in order to estimate how the reality differs from what was expected and provide correctives for the control signal, or "tuning" for the matrices A and B (using available techniques of extrapolation).
Thus, the following sets of information can be considered initially known:

S-states : $\{x(0),\ x(-1),\ x(-2),...\}$
P-parameters : $\{A(0),\ B(0),\ A(-1),\ B(-1),\ ...\}$
G-goal states : $\{x(k),\ x(k-1),...\}$.

The general form of a control-generating rule can be represented as follows

$$\text{IF } [P\wedge S\wedge G]^{*} \text{ -> THEN DO } U$$

where U -is a string $\{u(1),\ u(2),...,\ u(k-1),\ u(k)\}$,
*-means that the solution is Pareto-optimal.

Solution can be found by one of the selected search procedures. There is a set of optimum solutions, the cost difference between them can be found only at higher resolution.NDP is done a the following sequence of procedures:

SUBSTITUTE THE "OUT-OF-REACH" GOAL BY AN ACHIEVABLE GOAL
FIND A CONTROL-GENERATING RULE
IF THE CONTROL-GENERATING RULE IS NOT FOUND, APPLY SEARCH

SUBMIT THE SOLUTION TO THE NEXT LOWER LEVEL
IF THE SOLUTION IS NOT FOUND, REPORT THE PREDICAMENT TO THE NEXT
UPPER LEVEL

This approach is very general and produces the trajectories of motion in all cases: when the general rule exist and is known to the planning/control system, or when it does exist but is unknown. All algorithms of path-search in 2D binary world start with a statement IF THERE IS NO OBSTACLE BETWEEN THE ROBOT AND THE GOAL, GO DIRECTLY TO THE GOAL, ELSE DO THE SEARCH [29]. In fact, this rule must be proven by geometrical methods (i.e. at the higher level) or foung by a search after discretization and putting the problem to the lower level. This is the way of finding the trivial trajectories A and B in Figure 11,a.After the trajectories A and B are obtained as a result of the search procedure, the corresponding strings of commands can be stored for subsequent using them as solutions in corresponding rules [37].

In the case shown in Figure 11,b we have a less trivial situation. Indeed, the lower level search can lead us in a straightforward manner to the trajectory A. However, when the back-up motion is allowed, at least one additional trajectory should be added: a backing up motion B and then motion forward C. This new opportunity immensely reduces the productivity of search. One would probably prefer to first select one of these opportunities (A or B+C) at the higher level, and then compute the actual trajectory.

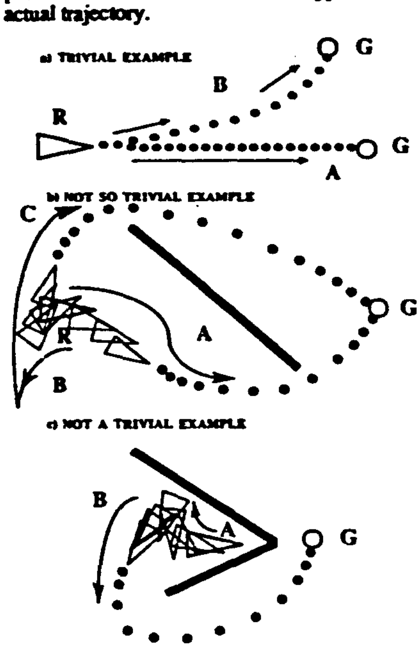

Figure 11. Examples of solving typical motions

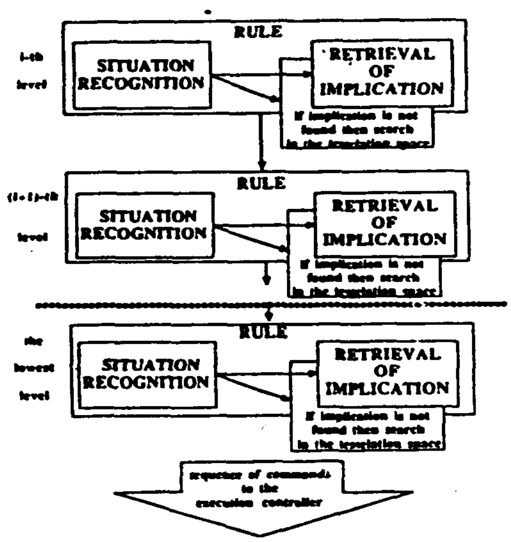

Figure 12. Sequence of Rules-Search Application in the Nested Hierarchical Controller

IF ROBOT IS BECOMING CLOSE TO AN OBSTACLE (DISTANCE IS LESS THAN D)
(ON THE RIGHT, OR ON THE LEFT)
THEN PUT MESSAGES TO PILOT AND EXECUTION CONTROLLER TO 1 PRIORITY
AND
IF DISTANCE IS LESS THAN D* AND MORE THAN D**
THEN REDUCE SPEED TO V* AND MAKE A TURN
(LEFT, RIGHT)
AND
IF DISTANCE IS LESS THAN D**
THEN STOP AND REPLAN

One can see that these rules do not refer to any specific global coordinates of the trajectory to be executed, and the local coordinates (position of the obstacle relative to the robot) are given with low accuracy using linguistic variables as it was done in [28]. Similar rules are formulated for a more subtle maneuver when after the local goal was determined in a particular situation, it turned out to be in an inconvenient location, e.g. the actual location of G^{**}_{i+2} (see Figure 6) cannot be achieved in a near-minimum time fashion by a simple continuation of the motion ahead.

IF G(X) IS LESS THAN R^{T}_{MIN}
AND G(Y) IS LESS THEN 2 R^{T}_{MIN}
AND ORIENTATION IS LESS THEN |90°|

THEN BACK-UP AND TURN IN THE DIRECTION OPPOSITE TO THE GOAL

(G(X) and G(Y) are the "x" and "y" coordinates of the local goal in a local reference frame, i.e. when the coordinate system is attached to the robot, R^T_{MIN} is the minimum radius of turn).

Tis rule allows to perform a nontrivial motion shown in Figure 11,c which reproduces a very anthropomorphic way of dealing with this problem.No wonder - it was introduced by a human "masrer".We do not have any experience in formulating rules like this automatically.It seems that using the human experience and intuition in formulating rules would be beneficial for dealing with predicaments arising during processes of operation.

The overall process of using the rules previously formulated and stores, and the search when the rule is not found in the list, is illustrated in Figure 12.

Learning of rules. Thus, if the rule is not on the list, the recommendation is obtained using the search in the state space, or in other words, by solving the automaton of representation, or by solving difference equations of representation via (say) enhanced dynamic programming [31, 33, 36]. The results of this search can be stored and later used as a rule. Then the number of rules will grow tremendously, and instead of spending time for search the system will spend time for browsing the lists of rules. Probably, there exist an optimum number of rules depending on the descriptive properties of the space as well as the characteristics of the computer system utilized in a specific intelligent robot.

On the other hand, if a group of rules can be generalized and pushed to the upper level of the hierarchy, then the overall time of search can be reduced. This generalization can be done as follows

DETERMINE FEATURES OF CONDITION PARTS OF RULES IN STORAGE
FIND SIMILARITIES AMONG THE FEATURES OF RULES IN STORAGE
GROUP THE CONDITION PART OF RULES IN STORAGE
GROUP THE CONSEQUENT PARTS OF RULES WITHIN THE GROUP OF RULES
WITH SIMILAR CONDITION PARTS
SUBSTITUTE ALL CONSEQUENT PART BY ONE (AVERAGE) SOLUTION
TO THE CORRESPONDING LIST OF RULES

The process of learning with new rules generation is illustrated in Figure 13. The stage of "generalization" can be performed so far only with "master" direct participation. Indeed, the control strings found as a response to the concrete situations should be stored but the subsequent analysis should be done by a "master" unless the reliable algorithms of generalization are developed.

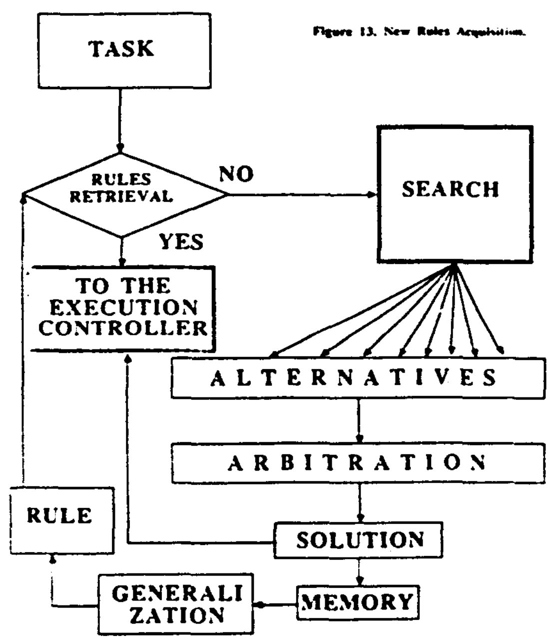

Figure 13. New Rules Acquisition.

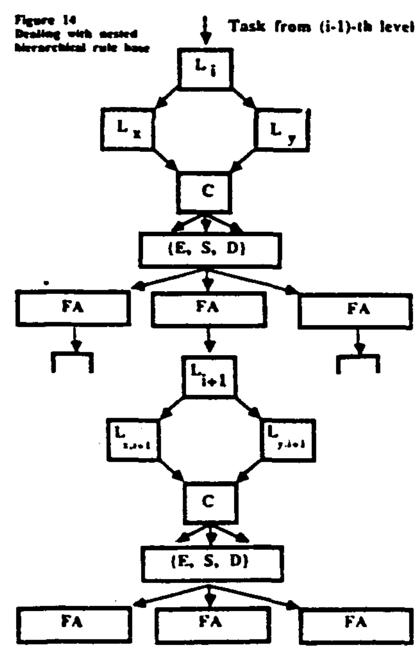

Figure 14 Dealing with nested hierarchical rule base

In the meantime, the other types of learning (primarily, inductive learning) can be excercised with no human involvement. They include memorizing the snapshots of the world, analysing the evolution of the snapshots, and computation of correctives for the matrices A and B as well as for the string of U.

Retrieval of rules. The first experience of operation of autonomous mobile robot Drexel Dune-Buggy has demonstrated that usin the lists of rules is inefficient, and the hierarchical structure increases the productivity of the rules retrieval. The structure is shown in Figure 14. It is based upon the theory presented in [38]. L_x is the language of conditions consisting of their vocabulary and the weighted relationships among them. L_y is the language of consequents also consisting of the vocabulary and the grammar of weighted relationships among the words of this vocabulary. The matching algorithm is based upon the grammar translator consisting of the weighted relationships among the words of L_x and L_y. Tme matching process C ends with obtaining several "best" machings, then evaluation, selection, and decomposition of the proper rule is done ({E,S,D}).

After decomposition is completed, focusing of attention should pe performed (FA), after this the search of the proper rule of the next level can be performed. The process is not fully automated since the process of focusing attention not always can be done with no master involvement.

IV. Conclusions

In existing systems the function of "master" is working in the DIM-area. This presumes that master's participation is expected to be required at the stages of

context selection,
arranging the blackboards for the specific context,
assignment of the search envelope,
supervised learning.

To be more specific his functions will include (but not limited to) the following activities:

1. Making available the "universe of discourse" knowledge.
2. Proper selection of the context subset.
3. Monitoring the processes of image recognition especially while sensor integration.
4. Final judgment on the space discretization (uniform, nonuniform,etc.) based upon master's opinion on what is "meaningful neighborhood".
5. Monitoring the process of cost-function assignment.
6. Monitoring the process of vocabulary determining for the adjacent levels.
7. Elimination of discrepancies in world representation at the adjacent levels.
8. Monitoring the assignment of the relations of nesting by generalization.
9. Determining the scope of attention according to the general assignment of the operation.
10. Final decision on the duration of retaining the stored information especially for the levels of ephemeral maps.
11. Determiniming the rules of the high level of generality
12. Arbitration among the multiple alternatives.
13. Direct dealing with the problem of the lower level if the upper level vocabulary is poor.
14. Participation in the processes of new rule recognition.
15. Doing the job of linguistical tesseletion: determining the groups of entities belonging to the same level of discretization.

Expectantly, the language for communication with the "master" will be different from the language to be utilized for dealing with the other robots of the cooperative team.

V.References

1. H. P. Birmingham, F. V. Taylor, "A Design Philosophy for Man-Machine Control Systems", Proc. of the IRE, Vol. XLII, No.12, December 1954

2. N.H. Mackworth, "Researches on the Measurement of Human Performance", from Medical Research Council Special Report Series 268, London, GB 1950, Published in *Selected Papers on Human Factors in the Design and Use of Control Systems*, ed. by H. Wallace Sinaiko, Dover, New York, 1961

3. Bejczy, A.K., "Smart Sensors for Smart Hands", in Progress in Austronautics and Aeronautics, Vol. 67, Publ. AIAA, New York, 1979

4. Bejczy, A.K., "Effect of Hand-Based Sensors on Manipulator Control Performance", in Mechanism and Machine Theory, Vol. 12, Pergamon Press, 1977

5. Bejczy, A.K., Control of Remote Manipulators, in Handbook of Industrial Robotics, ed. by S. Y. Nof, Wiley, New York, 1985

6. A. Meystel, "Planning in a Hierarchical Nested Controller for Autonomous Robots", Proc of the 25-th IEEE Conference on Decision and Control, Athens, Greece, 1986

7. A. Meystel, "Nested Hierarchical Controller for Intelligent Mobile Autonomous System", Proc. of the International Congress on Intelligent Autonomous Systems, Amsterdam, Netherlands, 1986

8. A. Meystel, "Nested Hierarchical Intelligent Module for Automatic Generation of Control Strategies", Proc. of the NATO International Advanced Research Workshop on Languages for Sensor-based Control in Robotics, Ed. by U. Rembold, K. Hormann, Karlsruhe, FRG, 1986

9. A. Meystel, "Hierarchical Algorithm for Suboptimum Trajectory Planning and Control", in Recent Trends in Robotics, eds. M. Jamshidi, J.Y.C. Luh, M. Shahinpoor, North-Holland Elsevier, New York, 1986

10. A. Meystel, "Autonomous Mobile Device: A Step In the Evolution", in Applications in Artificial Intelligence, Ed. by S. Andriole, Petrocelly Books, Princeton, NJ, 1985

11. A. Meystel, "Intelligent Control of a Multiactuator System", in IFAC Information Control Problems in Manufacturing Technology 1982, ed. by D.E. Hardt, Pergamon Press, Oxford, 1983

12. A. Meystel, G. Hoffman, E. Koch, T. Looney, J. McKisson, M. Roberts, "Anthropomorphical Control of Intelligent Mobile System", Proc. of IEEE Southeastcon-82, Destin, Fl, 1982

13. G. Giralt, R. Chatila, M. Vaisset, "An Integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots", In Robotics Research, Ed. by M. Brady and R. Paul, MIT Press, Cambridge, MA 1984 9. A.M. Parodi, J.J. Nitao, L.S. McTamaney, "An Intelligent System for Autonomous Vehicle", Proc. of IEEE Intl Conf. on Robotics and Automation, San-Francisco, CA 1986

14. A.M. Parodi, J.J. Nitao, L.S. McTamaney, "An Intelligent System for Autonomous Vehicle", Proc. of IEEE Intl Conf. on Robotics and Automation, San-Francisco, CA 1986

15. D.W. Payton, "An Architecture for Reflexive Autonomous Vehicle Control", Proc. of IEEE Intl Conf. on Robotics and Automation, San-Francisco, CA 1986

16. A. Meystel, "Knowledge-based Controller for Intelligent Mobile Robots", in Artificial Intelligence and Man-Machine Systems, ed. H. Winter, series "Lecture Notes in Control and Information Sciences, 80, Springer-Verlag, Berlin, 1986

17. G.N. Saridis, Self-Organizing Control of Stochastic Systems , Marcel-Dekker, New York, 1977

18. G.N. Saridis, "Toward the Realization of Intelligent Controls", Proseedings of IEEE, 67, August, 1979

19. G.N. Saridis, "Intelligent Robotic Control", IEEE Transactions on Automatic Control, Vol. AC-28, No. 5, 1983

20. G.N. Saridis, J.H. Graham, "Linguistic Decision Schemata for Intelligent Robots", Automatica, Vol. 20, NO 1, 1984

21. G.N. Saridis,"Foundations of the Theory of Intelligent Control", Proc. of the IEEE Workshop on Intelligent Control, Troy, NY, 1985

22. R. Bellman, Dynamic Programming, Princeton University Press, Princeton, NJ 1957

23. Y. Bar-Shalom, "Stochastic Dynamic Programming: Caution and Probing", IEEE Transactions on Automatic Control, Vol. AC-26, NO. 5, Oct. 1981

24. H. Herrlich, G.E. Strecker, Category Theory, Allyn and Bacon, Inc., Boston, 1973

25. A.N. Kolmogorov, S.V. Fomin, Introductory Real Analysis, Prentice Hall, Englewood Cliffs, NJ 1970

26. M.D. Davis, E.J. Weyuker, Computability, Complexity, and Languages, Academic Press, NY, 1983

28. C. Isik, A. Meystel, "Knowledge-based Pilot for an Intelligent Mobile Autonomous System", Proc. of the First Conference on Artificial Intelligence Applications", Denver, CO, 1984

29. E. Koch, C. Yeh, G. Hillel, A. Meystel, C. Isik, "Simulation of Path Planning For a System With Vision and Map Updating", Proc. of IEEE Int'l Conf. on Robotics and Automation. St. Louis, MO. 1985

30. R. Chavez, A. Meystel "Structure of Intelligence For An Autonomous Vehicle", Proc. of the IEEE Int'l Conf. on Robotics and Automation, Atlanta, GA, 1984,

31. A. Guez, A. Meystel, "Time -Optimal Path Planning and Hierarchical Control via Heuristically Enhanced Dynamic Programming: a Preliminary Analysis", Proc. of the Workshop on Intelligent Control, Troy, NY, 1985

32. P.E. Hart, N.J. Nilsson, B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum-Cost Paths", IEEE Transactions on Systems, Science, and Cybernetics, Vol. SSC-4, No. 2, July 1968

33. A.Meystel, A. Guez, G. Hillel, "Minimum Time Path Planning for a Robot", Proc. of the IEEE Conf. on Robotics and Automation, San Francisco, CA, 19867

34. P. Graglia, A. Meystel, "Minimum Time Path Planning in the Traversability Space of an Autonomous Robot" Proc. of the IEEE Symposium on Intelligent Control, Philadelphia, PA, 1987

35. D. Gaw, A. Meystel, "Minimum-Time Navigation of an Unmanned Mobile Robot in a 2 1/2 World with Obstacles", Proc. of IEEE Conf. on Robotic and Automation, San Francisco. CA, 1986

36. A. Meystel, Primer on Autonomous Mobility, Drexel University, Philadelphia, PA, 1986

37. R. Bhatt, D. Gaw, L. Venetsky, D. Lowing, A. Meystel, "Dynamic Motion Guidance and Trajectory Tracking for an Autonomous Vehicle", Proceedings of the IEEE Intl. Symposium on Intelligent Control, Philadelphia, PA, 1987

38. A. Meystel, "Purposeful Arrangements via Search in the Version Space", Proceedings of the IEEE Intl. Conference on Systems, Man, and Cybernetics, Halifax, Nova Scotia, Canada, 1984