6

Nested Hierarchical Control

A. Meystel

Drexel University, Philadelphia, PA 19104
e-mail: meysteam @ duvm. ocs. drexel. edu

Abstract

In this paper, the theoretical foundations are outlined of decision making in the class of control systems which allows for using nested representation, and nested algorithms of control processes. As a result, nested hierarchies of multiresolutional (multiscale, multigranular) control structures are generated. The core of the theory of nested hierarchical control is based upon a concept of nested hierarchical knowledge organization which enables efficient practice of design and control using nested search in the state

1. INTRODUCTION: OVERVIEW OF THE AREA.

Control hierarchies that came from the 60-s [1-3] were based on the idea of system partitioning. G. Saridis' conceptual snapshot of the situation in the area of hierarchical control [4] reveals some of the major features typical for the hierarchical control systems: controller at the top of the system, controls the process as a whole; controllers at the bottom control the subprocesses, the latter should be coordinated. On the other hand, controller at the top is imprecise, it deals with the process at the level of linguistic descriptions; controller in the middle is more precise, but it is still a fuzzy controller; controllers at the bottom have the required precision.

J. Albus noticed that the structure of a hierarchical controller is similar to the structure of brain functioning, and that the hierarchy is generated as a result of "task decomposition" [5]. G. Giralt, R. Sobek, and R. Chatila are applying the task decomposition to the problem of mobile robot control [6]. It becomes clear that a hierarchy of functioning evokes not only a need in hierarchical decomposition of tasks, but also a hierarchical decomposition of maps (representations). J. Albus outlines for the area of robotics [7] the structures of brain functioning/hierarchical control as the three interacting hierarchies of task decomposition, world model, and perception. Motivated by these developments A. Meystel [8] proposes a control architecture "Planner-Navigator-Pilot" for robots. This architecture is dominating the area in the 80-s [12-14, 17, 19, 20, 24].

G. Saridis arrives with the principle of increasing level of intelligence with reducing precision bottom-up in the hierarchies of control [9]. It becomes clear that there are some general properties of knowledge processing in the control hierarchies, and that these properties are not determined by the phenomenon of system partitioning: they rather imply partitioning of representation which happens at the highest levels by the laws of linguistics [10], at the middle levels by the laws of fuzzy control [11], and they allow for integration of upper level with the lower ones [12-14]. A hypothesis is proposed [15] that control commands can be obtained at all levels as a time-tagged hierarchies of actions (procedural knowledge) which can be obtained by a corresponding processing of the snapshots of the World (declarative knowledge). Different strategies of mathematically rigid controllers are proposed [16-17], and eventually, a sketch of the theory of nested hierarchical control appears in 1986 [18].

Known applications are related to the areas of autonomous and teleoperated robots [19-24, 27-32] as well as for the area of material processing [36, 37]. In the meantime, the structure of the theory is becoming more clear [25, 26, 33-35] as well as the problems that should be solved [38-41]. This paper is a further development of earlier papers [18, 26, 29, 30]. It formulates theoretical methods of design and control in systems which allow for multiresolutional world representation and nested decision making. Motion planning, and motion control which are usually treated separately, are becoming a continual process in this approach (joint planning-control systems).

The ideas of nested hierarchical (multiresolutional, multiscale, multigranular) control are deeply rooted within numerous efficient mechanisms of knowledge representation [43]. Hierarchies of 60-ties [1-3] were focused upon as an organizational tool, and M. Minsky's "frames" (1975) can be considered the first explicitly discussed generator of nested knowledge [54]. Broadly utilized in the practice of programming as a part of LISP, nesting became also an important tenet of the so called "entity-relational approach". B. Mandelbrot announced that the Nature as a whole is built upon "fractally hierarchical patterns" (see p. 93 in [55]). Mathematical treatment of nested representations was explored by H. Samet during 80-ties (collected in [56]). Nested Hierarchical (multiresolutional, multiscale, multigranular) representation has generated a rich flow of research results in the area of vision (see [57-60].

Nested Hierarchical algorithms were introduced in the area of computational mathematics as "domain decomposition", or "multigrid methods" [51-53]. Hierarchical Aggregation of Linear Systems with Multiple Time Scales was discussed in the paper of the same title [61]. Multiscale statistical signal processing was recommended in [62]. Recently, an effort to formulate a Mustiscale Systems Theory has been done [63, 64].

One can see that most of the research results are related to development of models of computer vision, and to the general signal processing. Among the early papers related directly to multiresolutional (multiscale) controllers we can mention only [18, 21, 65, 66]. Strong connection of nested hierarchies of representation was early appreciated by the researchers (see survey [67]). An assumption was proposed in [67] called the *time-scale separation hypothesis*, which stated that some of the motion trajectories can be considered independently. An important problem was raised about controlling a class of systems which is represented inadequately [44, 45]. The focus of this paper is is also on systems: with incomplete and inadequately representation. We will try to equally reflect the possibility of a general theory of Nested Hierarchical Control, as well as to reflect the kindred methods: domain decomposition, local techniques, etc. Both are required for the engineering practice of design and development.

This paper is organized as follows. We give a brief outline of the evolution in the area of multiresolutional (multiscale) controllers (Section 2). The *nested hierarchical control* structures are demonstrated, and the principles of control are recommended. A broad theoretical paradigm is proposed (Section 3) based on blending the approaches from knowledge engineering, algebra, and control theory, and it gives foundations for multiresolutional (multiscale) knowledge organization and processing. The principle of Nested Hierarchical Control is discussed as a part of the Autonomous Control Systems ((Section 4), and the algorithms of NHC are presented. The list of references reflects the areas of nested hierarchical control as well as multiresolutional (multiscale) information (knowledge) processing.

2. EVOLUTION OF THE MULTIRESOLUTIONAL CONTROL ARCHITECTURE

General Structure of the Controller. Any machine and/or technological process can be easily identified with Figure 1, a where the following three parts can be distinguished:

A-a source and a storage of the World Model: it contains all Knowledge necessary for modelling the operation as well as the means of communication to acquire this knowledge from the external source (say, an expert, or a collection of models).

B-a computer controller which contains all necessary means for processing information delivered by sensors from the machine, evaluate the situation, and compute plans and immediate commands for controlling the machine.It organizes and stores the newly arrived information, it identifies the entities of the World to be controlled, it compares the assignment with the current situation and outlines the output to be obtained in order to achieve the desired goal of operation.

C-the machine performing the process of interest with actuators that transform control commands into actions, and with sensors that inform the computer-controller about the process. Figure 1, b shows a simplified version of the diagram from Figure 1, a. W - is the World, or the process to be controlled, S - is a set of sensors, P - is a system for dealing with sensor information ("perception"), K- a system of Knowledge representation, interpretation, and analysis, P/C - is a subsystem for planning and control which determines the required course of actions, and A - is a system of actuators which introduce the desired changes into the World. This simplified version of the information flow structure will be called *a six-box diagram.*

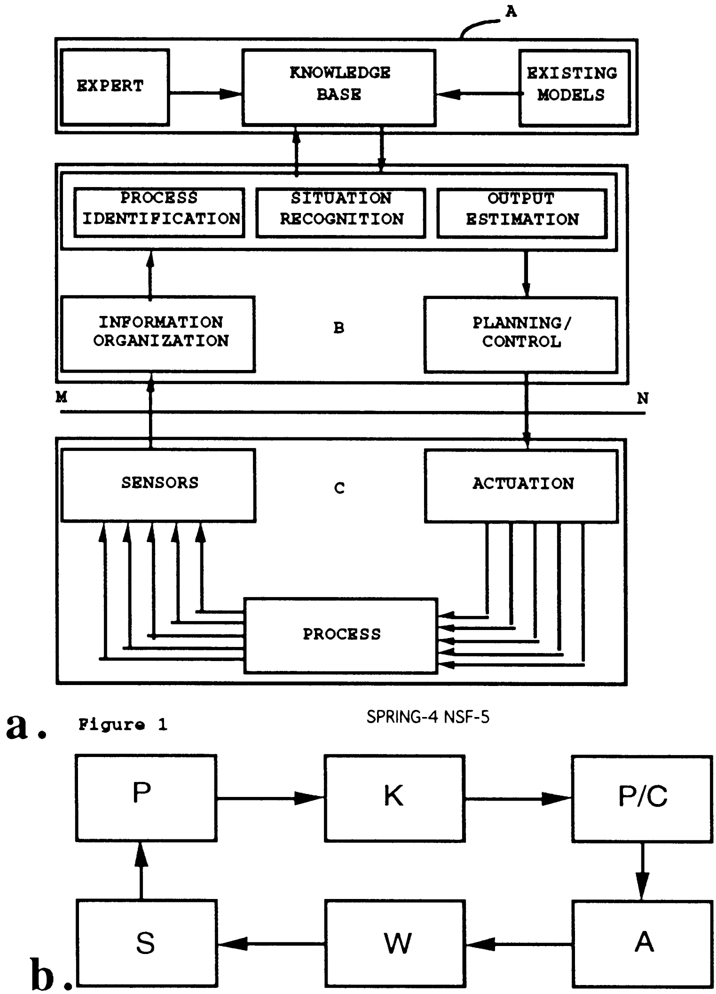


Figure 1.General Model of the Machine with its Computer-Controller.

A Case: Control of the Spray-Casting Machine. Let us consider a technological example illustrating that the system is perceived by a designer (or by a control engineer) at multiple levels of resolution. Manufacturing

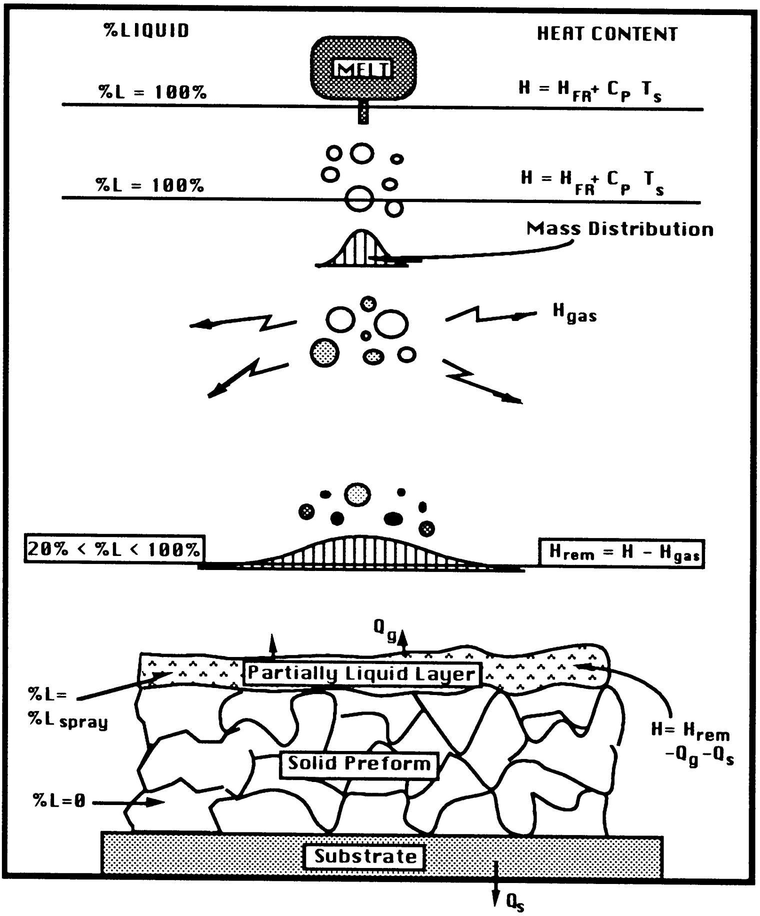


Figure 2. Physical description of the process to be controlled.

of sophisticated shapes can be done by using a so called *spray-casting*. Spray solidifies very quickly providing a high quality microstructure. With the substrate sophisticated motion, the shape of the growing part can be of high complexity. In Figure 2 the process of spray of metal is shown. Molten metal is contained in a sealed crucible at the top of the machine, and its temperature is being controlled.

The overpressure of inert gas above the molten metal can be controlled to change the "metal flow rate" during the process of spraying. The metal rapidly emerges from the ejection nozzle, it is atomized and thus transformed in a spray "cone". As a multiplicity of droplets it falls down on a substrate, which is being moved by a robotic arm. Droplets cool down during the flight (the rate of cooling can be controlled by the cooling gas pressure). As the droplets fall on the cold surfase, they solidify and gradually the required shape is being formed.

It is easy to distinguish at least two groups of the physical phenomena demonstrated in Figure 2. Firstly, there are such micro-phenomena like motion of every droplet starting with ejection and ending with its solidification. Obviously, the mechanical and thermal processes related to each droplet cannot be controlled with certainty: we cannot address each particular droplet. The statistics of droplets is an intermediate knowledge for considering a set of macro-phenomena which is the motion of the cone of the spray with its parameters, or the growth of the body of solidified metal with its parameters and characteristics.

In Figure 3 the actuators that are controlled to affect these processes are listed.The "overpressure" will affect the processes of ejection and transportation of metal to the substrate (Actuator #1). Another actuator (#2) affects the temperature

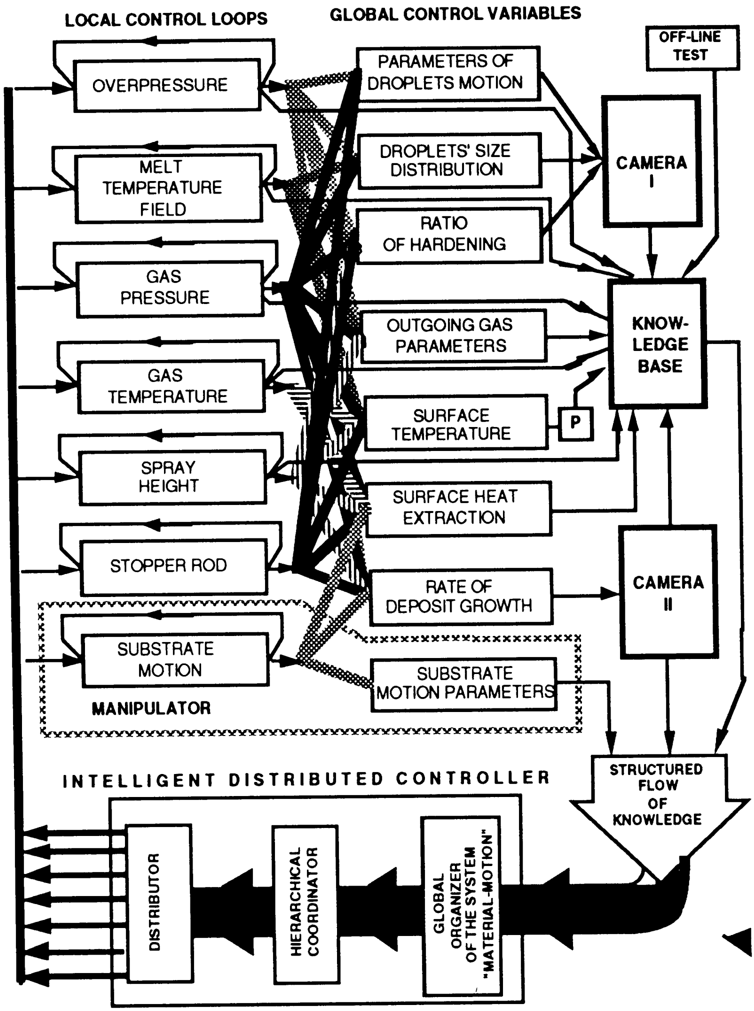


Figure 3. Structure of the controller for the spray-casting machine

of the metal ("super-heat"). The atomizing gas is being blown by Actuator #3 in order to produce proper atomization. Temperature of this gas can be transformed by Actuator #4. (Actuators #5 and #6 change the pressure and the temperature of the secondary gas which shapes and cools the spray cone).Spray height is changed by Actuator #7.The stopper rod opens the window through which the metal is being injected (Actuator #8). A group of actuators (#9 through #14) provides a sophisticated motion of the robotic arm holding the substrate.

The secondary gas determines the shape, focusing (width) and direction of the spray cone. Also it determines how quickly the particles will cool down during the flight. So the cooling gas velocity and temperature are becoming important. The motion of the substrate determines the final shape and the processes of cooling. The height of the spray determines the processes of cooling and determines the diameter of the bottom part of the spray, which affects the shaping of the preform. In the next column the list of global variables is shown (see Figure 3) which includes the parameters of droplet motion, droplet size distribution, ratio of hardening, outgoing gas parameters. It is clear that all of these variables belong to the inner processes, and part of them is reflected in their physical model.

The inner variables are shaping the output, which is demonstrated here as being collected within the knowledge base that represents the process. Most of the global variables can be judged upon only by indirect observations e.g. by video-cameras. It is clear from the structure of the controller demonstrated in Figure 3 that the actuators can be controlled by their individual PID controllers *if the assignment for them has been already computed by another level of control*. In our case, it is computed by the fuzzy logic controller which evaluates the required rough compensation response when the process deviates from the preplanned trajectory. Another control level should compute the plan of operation.

Thus the machine is being controlled by a three-level controller [36, 37]:

•**UPPER LEVEL: PLANNING OF THE TIME-PROFILES OF INPUT VARIABLES WHICH ARE SUPPOSED TO ENSURE THE DESIRABLE OUTPUT TIME-PROFILES.**

•**MIDDLE LEVEL: COMPUTING COMPENSATION CONTROL IF THE PROCESS DEVIATES FROM THE PREPLANNED INPUT/OUTPUT TIME-PROFILES AND/OR THERE ARE OTHER INDICATIONS THAT THE PROCESS SHOULD BE INTERFERED IN.**

•**LOWER LEVEL: EXECUTES THE PLANS AND COMPENSATIONS COMPUTED BY THE UPPER AND MIDDLE LEVELS, TRIES TO PROVIDE ITS ACCURACY.**

One can see that the concept of this controller is equivalent to the "PLANNER-NAVIGATOR-PILOT" concept known from [8, 19, 26, 39].

Phenomenological Hierarchy. Nested Hierarchical Control Architecture (NHCA) emerges from the concept of multiresolutional representation of control processes (Figure 4). Since each real process can be considered with different resolution (accuracy, threshold in representing details) a multiresolutional hierarchy of control loops can be intoduced. The overall process (the entity of the physical phenomenon of the process) can actually be considered a sequential-parallel connection of a multiplicity of subphenomena. For example,

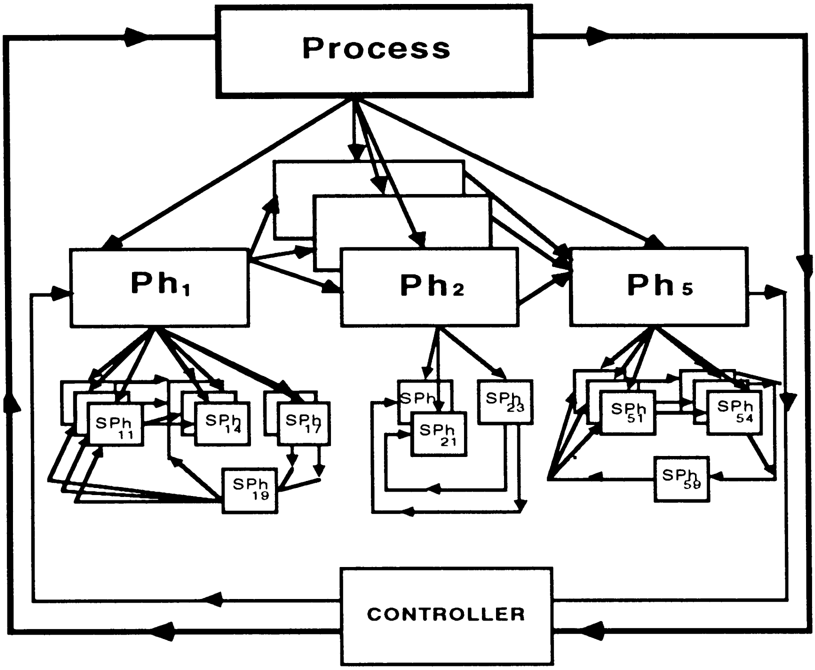


Figure 4. Phenomenological hierarchy of control processes.

the process of spray-casting contains subprocesses of a) heating-->b) ejection-->c) atomization-->d) spray formation-->e) spray flight with parallel cooling and partial solidification processes-->f) "landing" with parallel droplets sticking together, and shaping the final part because of simultaneously developing motion of the substrate-->g) gradual somidification of the part-->cooling.

On the other hand each of these subphenomena in turn can be considered a sequential-parallel connection of sub-subphenomena are actually available. For example: heating allows for describing non-homogeneity of the heating flow, fluid dynamics of the molten metal, generation of islands of different density within the molten metal, clogging the passageway for the metal precluding its ejection, etc. The behavior of spray can be described in the terms of statistics and thermal dynamics. The microstructure of "landing" processes includes the sub-subphenomena of bouncing off.

From Figure 4 one can see that sub-subprocesses are often presented in such vocabularies, using such models that their unification into the generalized model is impossible. A question can be raised: what is the mechanism of forming models of phenomenon out of multiplicity of the models for the subphenomenon, or vice versa if the models for the sub-phenomena is known. The answer cannot be given in a form of an algorithm or a universal rule. Multiresolutional representation is a very convenient tool of dealing with the multiple sub-phenomena. This leads to the Nested Hierarchical Control Architectures (NHCA).

Nested Hierarchical Control Architectures (NHCA). The NHCA diagram is shown in Figure 5. It is obtained as a result of redrawing Figure 1,b in such a way as to a) take in account the reality of Figure 4, and b) concentrate on the flows of control information in each level of the hierarchy. The following properties are characteristical for NHCA.

•Property 1. Computational independence of the resolutional levels. Each of the loops in Figure 5 can be considered and computed independently from others. Each of them describes the same control process with different accuracy and different time scale which entails the difference in the vocabularies of levels.

•*Property 2. Representation of different domains of the overall system processes resides at a level of resolution*. Since all loops are performing the same operation at different resolutions they are dealing with different subsets of the World (starting with the "small, fine grained, and quick" World at the bottom, and

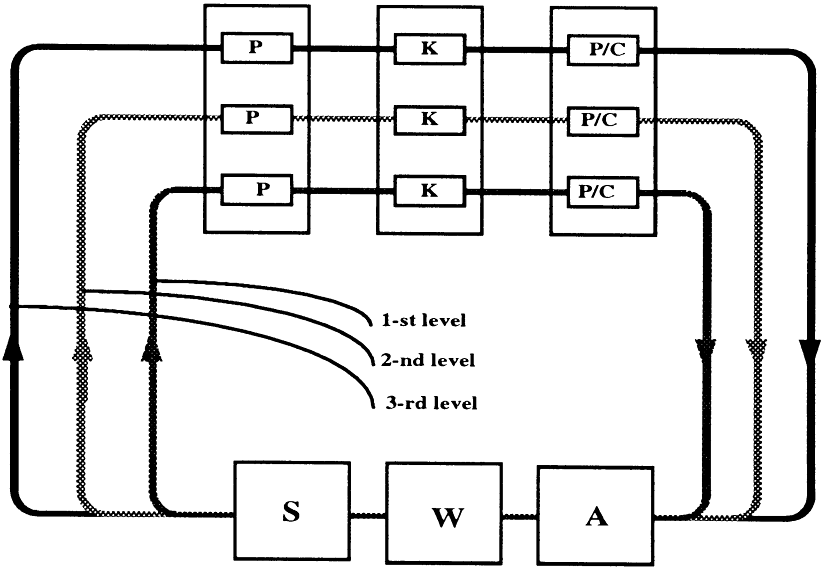


Figure 5. NHCA: each level has its own feedback loop: the lower levels are lumped into the “execution part”

ending with "large, coarse grained, and slow" World at the top).

•*Property 3. Correspondence between different levels of resolution different bands of frequencies within the overall process*. The resolution of the level is associated with the frequency of sampling which not only mean that the frequent sampling is associated with the higher accuracy of the processes representation, but also that the frequencies of the process which are lower than the frequency of the sampling are not likely to be reflected in the control processes of this level.

•*Property 4. Ability of the loops at different levels of resolution to integrate into the 6-box diagrams*. Loops are nested one into another (see Figure 5), the lower resolution loops presume a possibility of refining representation of their processes by using the higher resolution loops. Each of the loops contains Perception, Knowledge Representation, and Planning/control subsystems with the external World attached to them via Actuators and Sensors. In the meantime, they operate with different *scope of attention* : each process at higher resolution has a scope of attention narrower than the adjacent level of lower resolution.

•*Property 5. Correspondence between the upper and the lower parts of the 6-box diagram.* The upper part of the NHCA (P, K, P/C) corresponds to the lower part (S, W, A). The hardware realities of S-W-A are represented in computer architecture as P-K-P/C.

•*Property 6. Formation of the behavior of the system as a superposition of behaviors generated by the actions at each resolution level.* Action of the system is being generated simultaneously at several levels of resolution (granulation), e.g. if the teleoperated, or an autonomous mobile robot is considered. Then the list of levels bottom-up will be: 1) output motion level, (the lowest abstraction level, or the most accurate level), 2) maneuver level, 3) plan of navigation level, 4) scenario of operation level, 5) mission planning level (the highest abstraction, or the lowest resolution level).

•*Property 7. Similarity between the algorithms of behavior generation at all levels.* All levels execute a particular (pertaining to the level) algorithm of finding the best set of activities (control trajectory); each higher level constitutes the prediction for the each lower level. At each level, the action generating algorithms should perform: ASSIGNMENT GENERATION, PLANNING, PLANT INVERSE, DECOMPOSITION, COMPENSATION, and EXECUTION COMMAND GENERATION.

•*Property 8. Evolution of the hierarchy of representation from the linguistic one at the top to the analytical one at the bottom.* Several levels of planning/control processes presume a nested system of representations which can be analytical at the level of high resolution, and linguistic (knowledge based, or rule based) at the level of low resolution.

From the above properties, one can see that the process of control can be visualized as if an imaginary little robot was controlled from one location to another, capable of avoiding obstacles, or the disallowed zones, preferring low cost areas to the high cost areas, accelerating and decellerating, etc. Thus in each machine and/or process, the levels of multiresolutional control can be similar to what we stated for the mobile robot: 1) output motion level, (the lowest abstraction level, or the most accurate level), 2) maneuver level, 3) plan of navigation level, 4) scenario of operation level, 5) mission planning level (the highest abstraction, or the lowest resolution level). As one can see, this list fully apply to the general problem of *traveling within the state space.*

Before we are able to treat the system in Figure 5 as a system of control we will make several transformations. From Figure 5 one can see that all three control loops merge in order to enter the system to be controlled while dealing with processes of different resolution at different frequency. This allows for a conceptual leap: these three control loops can be considered independent loops (Figure 6, a). The Word (or the Machine) performs the superposition of the control loops working simultaneously: they can be mutually dependent, or independent, the nature of superposition does not change. At the next step the World is being decomposed into three different submodels each working within its own loop: the reality of the system to be controlled can be visualized as if three separate subsystems exist to match their control levels (Figure 6,b).

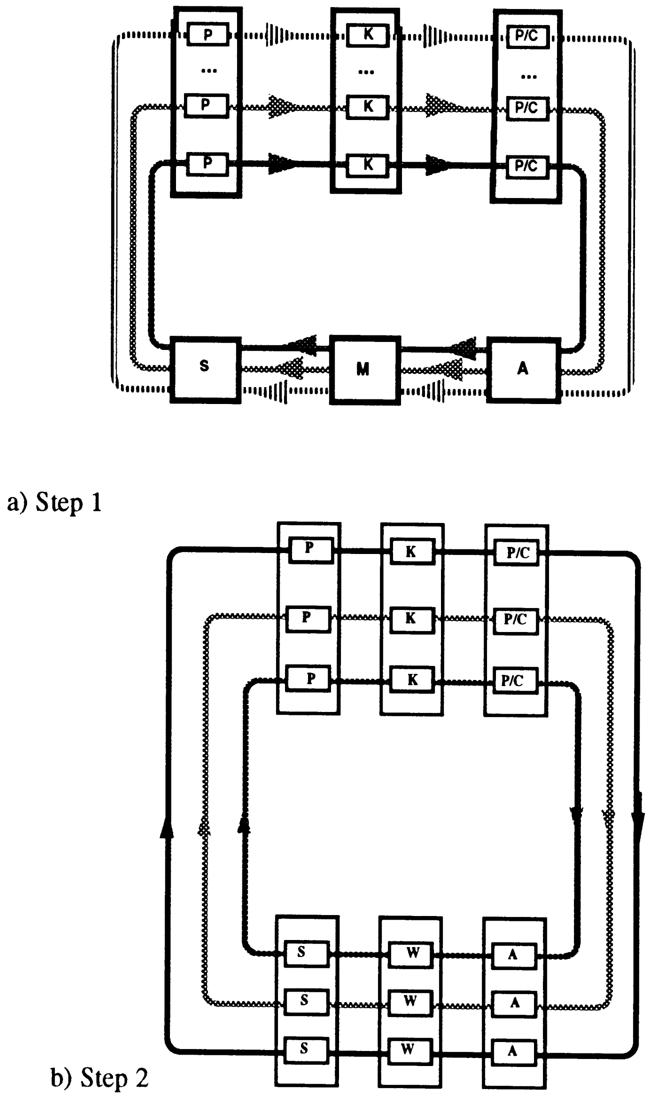


Figure 6. Multiloop Multiresolutional Controller

The example with autonomous mobile robot has a fundamental significance for the approach to design and control NHCA. If we consider a state space of the particular process (however complicated this process could be) the goal of control can be formulated as arrival from the initial point (state) to the final point (state) in this space. Thus, the moving point can be identified with an

It is a convenient way to model the World and to arrange for the computer controller as if they can be "component-to-component" mapping one into another. After the Step 2 is done we are able to deal with several World Models (as many as we have levels of resolution in NHCA). Each WM actually exists for the observer and interpreter associated with a particular level of resolution. The problem usually includes coordinating their actions as to optimize the process of goal achievement (see [42]) .

However, very often we are dealing with a special case of a single actuator system. A typical hierarchical control system allows for a tree-decomposition which leads to a tree-hierarchy exemplified in Figure 7,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 stem-hierarchy of decision-makers in order to be properly controlled (e.g. shown in Figure 7,b).

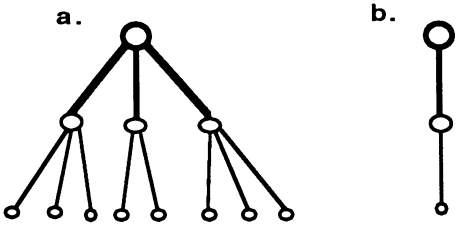


Figure 7. On comparison between (a) tree-hierarchy, and (b) stem-hierarchy.

3 . STRATEGIES OF NESTED CONTROL: GENERATION OF A NESTED HIERARCHY

Decision making procedures of planning-control. 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 8,a):

-Task: the goal G is to be achieved from the Starting Position SP, (and conditions to be satisfied including the parametrical constraints, the form of the cost function, its value, or its changes).

-Description of the "exosystem", or map of the world (M) including numerous items of information to be taken in account during the process of control; map of the world is often incomplete, sometimes, it is deceptive.

-Current information (I) is the information set about the vicinity of the working point delivered by the sensors in thevery beginning of the process of control, and continuing to be delivered during the process of ACS operation.

The process of control is to be described by the trajectory of "working point" moving in the state space. 6The processes within the controller are illustrated in Figure 8,b.

The part PT* of the overall planned trajectory PT, can be determined with higher accuracy. Our selection of PT can be changed in the future if the input

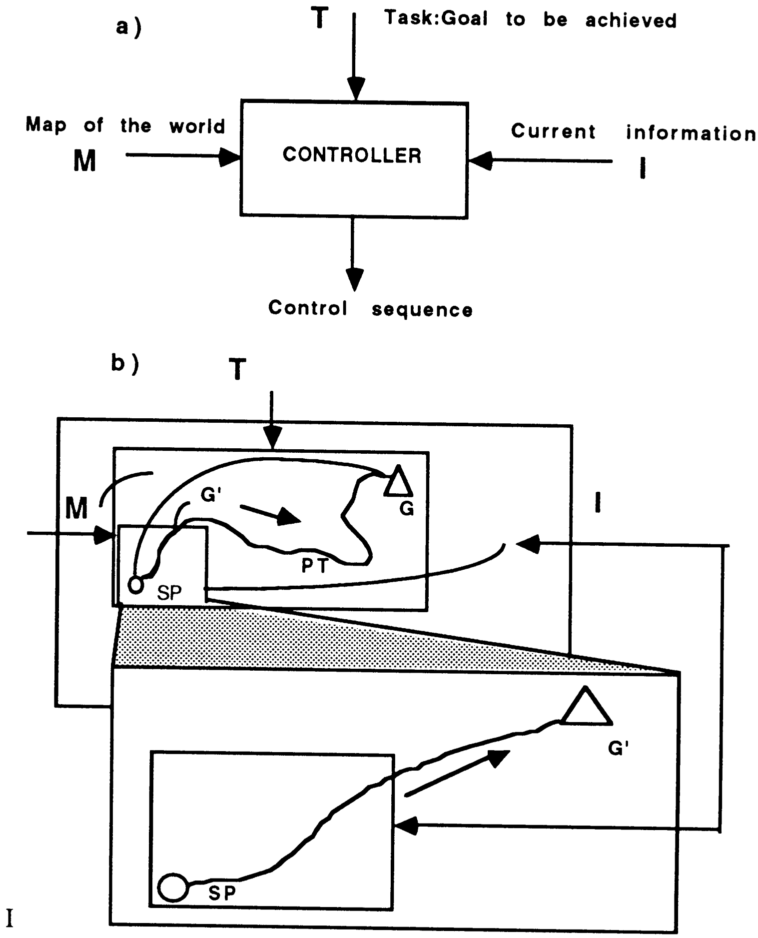


Figure 8. Controller and its inner off-line processes of decision making

updates the map M in the way that PT will not be the best trajectory. However, the PT* part of the plan won't be changed: no new information is expected. This includes the following components of the process of planning/control.

Planning/Control Algorithm.

1.Finding the optimum plan PT based upon the map M, ant the task formulation (SP, G, cost function, constraints) is done as follows:

•Search for the alternatives of PT.

•Comparison of the alternatives found.

•Selection of the preferable alternative accepted as a plan to be executed).

2.Updating the map information M in the vicinity of SP by using the sensor information I.

•Analysis and interpretation of the set I.

•Comparison between M and I.

- •Deciding upon required changes.
- •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.

- •Determining the subgoal G' (e.g. as a point of intersection of PT and the boundaries of updated part of the map).
- •Finding the optimun plan PT* (repetition of procedures P.1).

4. Track (follow) the optimum path PT*.

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

Both planning system and execution controller are supposed to work together to solve the tracking problem. The similarity between our "positioning control" and "tracking the target", is becoming even more noticeable after we realize that the tracking trajectory is being constantly recomputed during the process of theplan computation.

Nested Hierarchical Information Refinement during the Decision Making. The on-line process of consecutive information refinement is shown in Figure 9. At the top of the diagram the subgoal G'_i is found. Let us consider finding the refined plan PT* as a separate problem in which the new refined information I_{i+1} can be delivered for a part of the map M_{i+1} in the vicinity of the initial point SP. Obviously, this consecutive process generate a nested hierarchy of computations which is characterized by an important precondition for each level: the motion starts if the subgoal is determined at the upper level, and as the new information updates map in some vicinity, the new subgoal should be determined for the lower level (level of higher resolution) as a result of the post-motion path planning at a given level.

The best planned trajectory can be considered a predicted[1] trajectory. At the each level of the system shown in Figure 9, the circled 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. Within the plan assigned by the upper level, *a plan* is being refined, and the rest is becoming just a prediction.Since part of PT is cosidered as a prediction anyway, and the corresponding information is to be updated in the future. This generates such topics as storing the alternatives of plan, evaluation of predictions by some probability measure, and synthesis of contingency plans.

Generalized Controller. The problem of Generalized Controller design is illustrated by diagram Figure 10,a. Generalized Controller can be proven to consist of two major parts (Figure 10,b): Open-Loop-Controller (OLC), or

[1] Prediction is defined as declaration of the belief about the future events. Predictor in the system of control is a device which evaluates the *expected* future values of variables. Whether these variables are controlled, on uncontrolled, it does not matter as far as prediction is concerned. Thus, the result of prediction can be used to compute (in advance) the feedforward control as well as the compensation required. When the prediction coincides with the desired trajectory (highly probable prediction) it becomes a *plan*. If the value of probability of prediction is low, we cannot make it a plan, we must increase its probability.

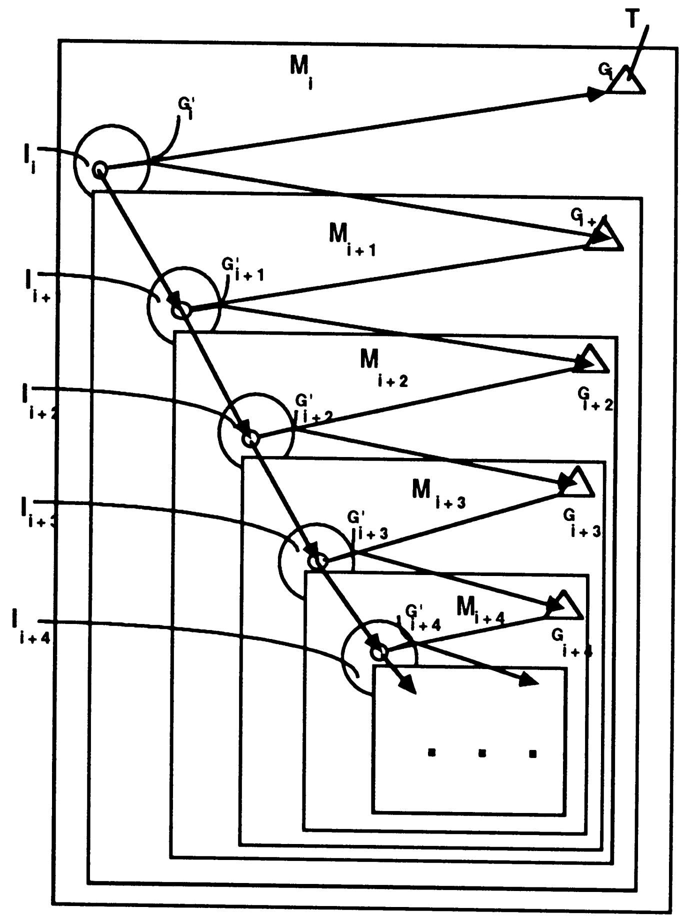


Figure 9. Nested hierarchical refinement of information during decision making

a *feedforward controller*, and Closed-Loop-Controller (CLC), or a *feedback compensation controller*. OLC is designed to generate a control input applied to the *plant* (G) to be controlled. Operation of planning (S) finds the output trajectory leading to the desirable final state with minimum cost of this operation. The only way to determine the *required input* to the plant is to construct the *inverse* (G^{-1}) of the plant and then apply to the input terminals of G^{-1} the *desired output* of the plant. Then, at the output terminals of G^{-1} the required input to the plant will be obtained, so that $G \bullet G^{-1} = I$, and whatever you submit to the input (desirable output trajectory) must appear at the output (actual output trajectories) to the degree of accuracy of our knowledge of G.

Certainly, the computational structure G^{-1} is obtained in assumption that the model G of the plant is known adequately which presumes knowing of the external environment which never is the case. Thus, when the computed required input is actually applied one should carefully compare the actual output with the desired output, and the difference (error) use as an input to another computational structure: compensating feedback which forms immediately the feedback compensation loop. It has been proven that in order to minimize the output error CLC should be also based on the G^{-1} computational structure.(The following subtleties should be taken in account before one uses these recommendations: the computational structure G^{-1} should be determined only after G is stabilized.).

Planning includes the procedures of search both for the desired output as well as for the input which should be applied to the plant when the desired output of the plant is given. Another term for this input is the *feedforward control*. If we compare the real output with the desired output, the error of OLC can be found and the compensation can be computed and applied. The output of the feedback loop we will call *compensation*. Compensation feedback controller we will call *closed loop control*, or CLC. It is clear that planning can be done off-line as well as on-line, in advance as well as in real-time. Notations of Figure 10 are related to the arbitrary level of any control system: T_{i+1}-task from the level above, G-the transfer function of the plant P_i (stabilized), T_i-task for the plant P_i, this task can be obtained from the task of the level above T_{i+1} by applying the operator of planning/control P/C= $S*G^{-1}*F$, where S is the generator of the string of input commands, G^{-1} is the inverse of the plant's transfer function, F- is the feedback operator (compensation). The plant of a level together with the planning control operator P/C is considered plant of the upper level (Pi+1).

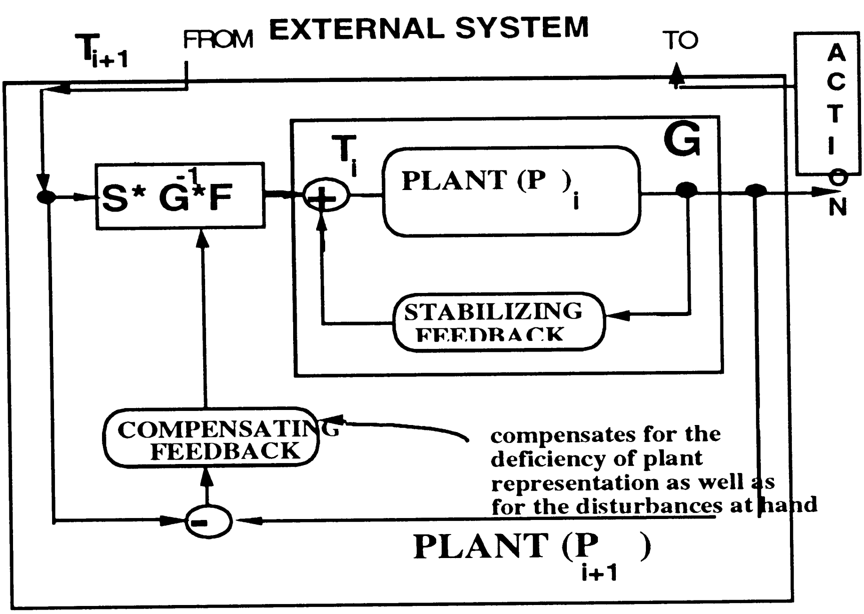


Figure 10. Realization of the Generalized Controller

Universe of the Trajectory Generator: 2-nd level. The Execution Controller, or the 1-st level of control (Figure 10) is a machine for producing sequences of preassigned output states. The next adjacent level above, or a Trajectory Generator (2-nd level) has at its input a *trajectory of motion required to perform a particular maneuver* (TMPM). The maneuver is understood as a recognizable sequence of evementary trajectories[2]. Then the whole diagram shown in Figure 10 (the 1-st level) is considered "The Machine for Executing TMPM" (e.g. this is a set of PID Execution Controllers for turning the all four wheels of our robot: all operators of CLC for G_1 are assumed to be Multi-Input-Multi-Output, or MIMO). MIMO TMPM Machine (shown in Figure 11,a for a particular example of application) has at its output not a motion of a single wheel, and not a set of motions of the four wheels but a trajectory of motion of the robot during some unspecified particular maneuver which is unknown at this level. In other words, these are the *actual time profiles* of speed, position, and orientation of the robot. In order to receive this output the desired TMPM has been obtained using $(S*G^{-1}*F)_1$ operator.

OLC is not expected to work properly because the actual load torques are not equal to the expected ones, a number of nonlinearities contaminate the picture such as nonlinearity of the friction, slipping factor between the wheels and the ground, and high order components of the Taylor expansion that were neglected in the model of transmission. Thus, the actual time-profiles of position, speed and orientation differ from expected ones, and CLC is needed for the error compensation.

The maneuver controller has at its input a call for a particular maneuver (Figure 11, b). MIMO Maneuver Generating Machine has at its output not a trajectory of robot motion as a time profile but rather a description of some particular maneuver which is a part of the motion schedule unknown at this level. In other words, the input is presented as a verbal (logical) description in the terms of standardized behavior which is expected from this robot. This OLC also is not expected to work properly because actual environment differ from the typical environment presented in the Look-up Table of the available TMPM goals. Thus, the actual sequence of the maneuver primitives can differ from expected ones. This difference is being measured and used to generate the CLC part of the controller at this level. compensates for the deficiency of plant representation as well as for such disturbances as "terrain surface on the left is of different quality than expected".

[2] Two new terms are used in this phrase: “recognizable”, and “elementary trajectories”. The latter term implies that there exist a vocabulary of elementary trajectories so that each real trajectory can be described as a concatenation of “elementary” ones. Some classes of these concatenations can be considered entities, they are used more often, we refer to them in decribing functioning of the machine (right turn, K-turn, etc.). These are entities of the next level of (lower) resolution, we recognize them as “maneuvers”.

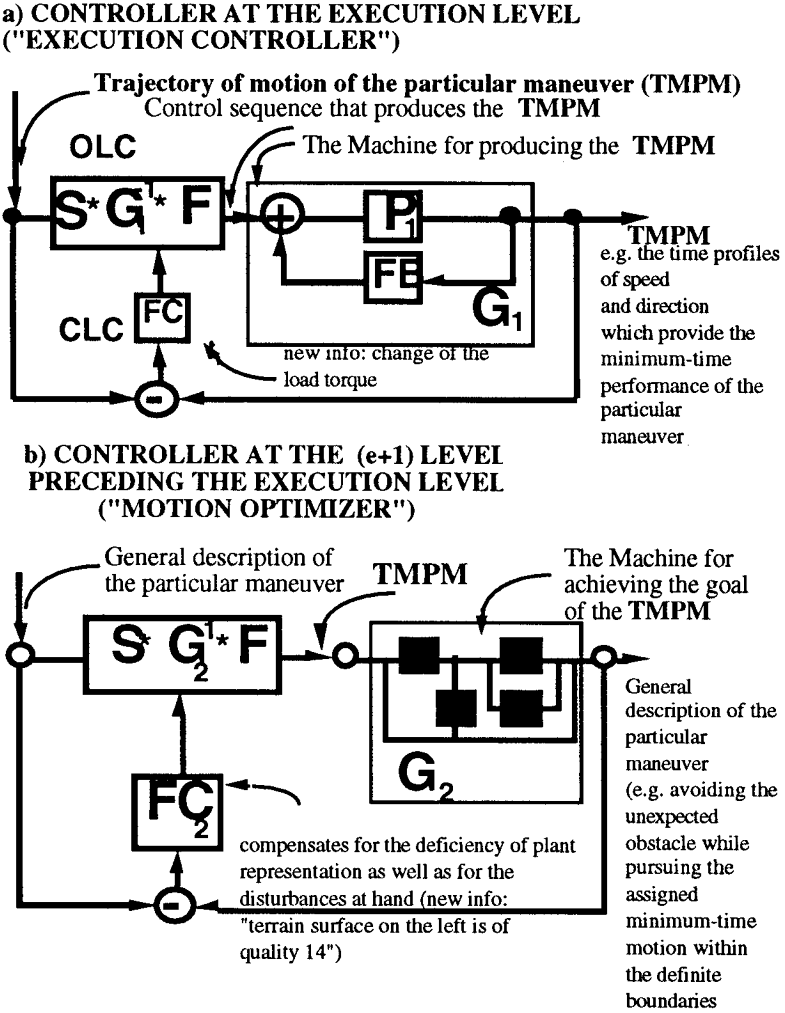


Figure 11. Controller at levels : a) trajectory generation, b) maneuver generation

Representation of the control problem. The control problem in nested hierarchies can be defined as follows. First, the dynamical system can be represented as a formal structure (e.g. mathematical, linguistical, or otherwise symbolic structure) or a “knowledge base” structure representing relationships between the states "x" and their rates of change "x'"

$$KB[\ X(t),\ X'(t),...],\ \ t=t_0,t_1,...,t_f, \tag{1}$$

where $X(t)$ and $X'(t)$ are the sets of time profiles of the evaluations for the variable and its rate of change in any available form. Time profiles are the strings of values of the variables and their rate of change directly stored (or otherwise computable by supporting procedures) for the whole duration of the process of interest starting with initial time t_0 and ending with the final time t_f .The knowledge base which is denoted symbolically as the structure (1) can be accepted in a form of difference, or differential equations, in a form of set of logical statements, or any another form assuming formal (and computer) manipulations (see in [46] an illustration of this statement). Here, we would like to stress the fact that the form (1) should be understood in a much broader sence than just an analytical form; it can be an abbreviation for any array of coded knowledge about the world realities denoted within (1).

The structure (1) should be supplimented by information on the state inequality constraints which in turn, depend on the actual state and the result of measurements and processes of recognition and interpretation

$$X_c = \{x_{ci}(x,x',...,t): \| x_{ci}(x,x',...,t \| \le x_{mi},\ i=1,2,...,k\}, \qquad (2)$$
$$X'_c = \{x'_{ci}(x,x',...,t): \| x'_{ci}(x,x',...,t \| \le x'_{mi},\ i=1,2,...,k\},$$

A subset of variables $U(x,t)=X_{inp}(t)\supset X(t)$ is considered to be controlled independently and is called *controls*. The structures (1) and (2) should also be supplemented by the information on the state dependent admissible control set

$$U_A=\{u_i(x,t): \| u_i(x,t) \| \le u_{mi},\ i=1,2,...,n\}, \qquad (3)$$

and on the cost-functional

$$J= \int_0^{t_f} L[x(t),x'(t),...,t]\, dt \qquad (4)$$

which characterizes the final cost of the process, and is supposed to be properly interpreted depending on the situation. For example, $L[x(t),x'(t),...,t]$ should be understood as a sub-knowledge base storing (or computing) the cost of being $x_i(t),x'_i(t),...,t_i$ at a particular moment of time "i " which allows for computing the cumulative cost of the process.The control problem is formulated as follows: "for a given map of the state-space[3] , for a given "initial point" , and "final point" (goal) of the motion:

1. declare part of the variables x(t) to be the output variables $X(t)...X_{out}(t)=y(t)$;

2. find the desirable output is proposed $y^*(t)$ which is called the *output plan*; the time profiles of the output vector components (*output plans*) can be found from the knowledge of the "starting position" $y_0(t)$ and final goal $y_f(t)$ using operator denoted S in the previous subsection; the tme profiles of the control vector components can be found from the output plans and the inverse transfer function of the plant denoted G^{-1} in the previous subsection.

3. find the *open-loop*, or *feedforward* control vector $u(t)$, or the *input plan* which minimizes, maximizes, or keeps within some inequality bounds the

[3] Since the cost is different of traversing units of the state space in different parts of the map, we will call it a "**variable-cost-map**".

value of J; Since neither S, nor G^{-1} are perfect, the real y(t) will differ from the desired y*(t) , and the difference Δ=y(t)-y*(t) should be compensated by the operator of feedback compensation F'.

We will consider here only additive law of compensation other laws can be considered too, it does not affect the substance of our presentation. So, in order to have the control problem solved, the structures (1)-(4) should be supplemented by the following structures

-for the compensated control:

$$u(x,t)=u^*(x,t)+F\bullet\Delta \quad (5)$$

-for the planned control:

$$u^*(s)= G^{-1*}\bullet y^*(s) \quad (6)$$

-for the error of control (deviation from the plan):

$$\Delta=y(t)-y^*(t) \quad (7)$$

-for the plan:

$$y^*(t)=S[y_0, y_0'; y_f, y_f'; J]. \quad (8)$$

For simplicity we will not pay too much attention at this stage to the question about the nature of the operator S and computational algorithms which are required for (8). Later we discuss it at length. At this point it would be instructive to assume temporarily that is given in advance: say, a problem of "tracking" is being solved, and the trajectory to be tracked is given externally. It is interesting to see whether we are adequately equipped to represent the problem of control, and to execute it if solved.

The system of structures (1)-(8) is assumed for a particular accuracy of represemtation. In the previous sections we saw that accuracy (level of generalization) was critical for determining the vocabulary of the level, therefore the number of variables and their contents was determined by the accuracy of representation too. Thus, we can expect that each resolution level will entail its own system of structures (1)-(8). In this paper, we are interested in finding how the systems (1)-(8) from all levels are related to each other.

The system of representation of the available (as well as of the required) information, is looming in the above considerations. Structures (1)-(8) built for real examples, strongly depend on the accuracy of representation, i.e. on the accuracy of the assumed processes of sampling, digitization, etc., and then, decoding, interpreting, storing, and executing the information. It is clear that the quality of representation should allow for planning, and executing the motion, and on the other side, to distinguish the difference Δ between the motion we were able to execute and the plan of this motion which was cood enough before the process started.

Here we have a dual situation. On one hand we are still within the realm of *representation for control* (1)-(4). However, when the complexity of Plant and World is growing, the information structures becomes very complicated and its computational complexity can grow inpermissibly. Thus, when we try to find a controller, say a feedback controller, the devices for receiving and interpreting information about the states and outputs ("perception") are becoming inpermissibly complicated, and the controller is becoming non-relilizable as a technological object: too cumbersome, too complex, too unreliable, probably, too expensive. This entails substantial problems by itself: special "knowledge representation" subsystems are required for perception, for knowledge base, and for the subsystem of planning/control.

Nested Hierarchical Production System: Levels of Abstraction. Even in its initial form, the controller at a level can be considered as a production system where the data-base (or the knowledge-base for declarative knowledge) is the initial structure (1)-(4) including the task, requirements of the concrete cost-functional minimization, constraints and the comments on dealing with them, the rule-base (or the knowledge-base for procedural knowledge) is a structure (5)-(8), containing the variety of algorithms for solving our problem in different cases. The third component of production system, or *control* (in AI sense) or *meta-rules* as we would prefer to call it, includes the premises that allow us to apply the rules: we can expect that as a meta-knowledge, the metarules should be sought for at the adjacent level of resolution above the level of consideration where the generalized information is contained.

Certainly, this can be repeated about each resolution level. The nested hierarchical production system for NHC can be introduced in a form compatible with Figure 5). For the example with autonomous mobile robot the lowest level, or the database includes all available deterministic information such as equations of motion, the updated map of the World, location of the robor, and the goal location.

Control Strategies contemplated for NHC. Search is a conventional technique of finding solution in the production system. In a particular case when a tree or a string should be found, a number of search algorithms is being used. One of the very efficient search-algorithms, "best-first" algorithm of search [49] is based upon the following sequence of operations:

Search Algorithm (general scheme) for finding minimum cost path on the graph

1. Define initial and final nodes of search.
2. Determine "successors" from the initial node which are to be considered as a next (after the initial) standpoint.
3. Determine cost for all of the successors.
4. Select the minimum cost successor as a new initial point.
5. Loop to the step 2.

The cost of each of the successors can be tetermined as a clear cumulative cost of achieving this particular successor from the initial point.

$$C_f = C_g + C_h \text{ , (or f=g+h as in [49])} \qquad (9)$$

where C_g- is the cost from the initial node to the one of the set of generated nodes-candidates, C_h-is an evaluation of the cost from the node-candidate to the goal.

It was shown, that when no additional information is available, one should determine the minimum possible value of distance between the candidate node, and the goal, using the accepted metric of the space of the search. This strategy leads efficiently to the optimum solution.

In this case the algorithm is called The Dijkstra Algorithm, and it propagates in all directions in the state space. The Dijkstra algorithm tends to check all possible paths before it selects the best one. A heuristic can be introduce which prunes the number of trajectories explored by including in the expression for the cost one additional component. In this algorithm (it is called A*-algorithm,the cost is computed as a sum of "clear" cost of achieving (or as a total relevance with) the successor, and of "less clear" cost of moving from the successor to the final node (or relevance between the successor and the final node). The A*-search propagates not as broadly as the Dijkstra-search, it tends to be "attracted" by the goal.

Search for a trajectory satisfying numerous constraints, and minimizing a cost-functional invokes variational methods, and ascends to dynamic programming (DP) as shown in [48]. The latter is repelling because of the well known "curse of dimensionality". Many efforts to apply DP were obstructed by subtle computational "glitches". Nevertheless, when not overestimated DP seems to be the most appropriate and perspective method because of the following considerations:

1) most of the systems we are dealing with in the NHC area, are substantially nonlinear, coupled, and cumbersome ones: off-line precomputation of table-look-up would be expected for control of such a system anyway;

2) we can consider DP as an idea of synthesizing trajectories in the state space, as an idea of a graph-search [48]; this allows for enhancement DP by a number of heuristical methods which are intended to make the algorithms more computationally efficient (e.g. [16, 17]).

Predecessors of NHC Algorithms. If a mapping F: $R_{\Delta 1}{}^{n} \rightarrow R_{\Delta 2}{}^{n}$ is given, where Δ_1 and Δ_2 are minimum errors registered (accuracies, resolutions of representation) in the first and in the second spaces, and if $\Delta_1 > \Delta_2$, and if F(x)=0 is to be solved consecutively in $R_{\Delta 1}{}^{n}$ and in $R_{\Delta 2}{}^{n}$, an initial solution X° found in $R_{\Delta 1}{}^{n}$ will be in sufficient closeness to the final solution in $R_{\Delta 2}{}^{n}$ so that if we continue this process in the spaces with smaller minimum errors ($\Delta_3 > \Delta_4 > ... > \Delta_{min}$)the computation process converges. It is clear from the above that this will give a nested hierarchy of optimum solutions which will be "enclosed" one into another. (The crucial condition of this is the mapping F: $R_{\Delta 1}{}^{n} \rightarrow R_{\Delta 2}{}^{n}$ which sometimes is difficult to provide). This consideration is one of the intuitive premises behind the method of "Nested Dynamic Programming" (NDP), or method of "nested consecutive refinement" which was proposed recently for systems of nested hierarchical control [16-19, 21-23, 25-28]. Similar concept was contemplated earlier for increasing efficiency of dynamic programming [50].

Similarities exist also between NDP and the method of "small parameter" where the systems

$$x' = f(x, \varepsilon u, t) \text{ or } x' = \varepsilon f(x,u,t) \qquad (10)$$

are considered instead of x'=f(x,u,t) [61]. Starting with a weakly controlled system and obtaining optimum control we are changing ε gradually from $\varepsilon \leq 1$ to 1, using optimum control for previous ε as an initial approximation. NDP can be considered an extention of continuation methods.

Multigrid methods are also employing the ideas of consecutive refinement [51-53]. The solution is initially found by numerical solution of a complicated differential equation at a low resolution (coarse granularity, or with a low resolution grid). Then a vicinity of this solution is being determined and only in this vicinity, a new finer grid is built, and the problem is being solved again.

4. DEVELOPMENT OF NHC-ALGORITHMS.

Extention of the Bellman's optimality principle. The principle of optimality of Bellman is 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 [47]. Three types of inclusion participate in the nested algorithms of dynamic programming: by generalization (g), by focus of attention (fa), and by time (t). The last points to the fact that the time sequence of information is nested, i.e. each next contains its predecessor.

Bar-Shalom has shown [47] that in the case of incompletely observed Markov process, stochastic dynamic programming can be applied.

Nesting of optimum decisions. Optimum decisions computed for different levels are nested if they are found under cost-functionals which constitute a nested system. (In particular, minimum-time controls found at different levels of resolution, are nested). 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 NHC.

Nested Hierarchical Search in the State Space. The problem of motion planning was given substantial attention in the literature on AI and robotics. However, problems of optimum planning (or optimum navigation, or optimum guidance) as well as optimum control until now do not have consistent and tractable solutions. Either solution is consistent or tractable but never both. Motion planning is frequently understood in the context of "solvability" of the problems of positioning or moving the object rather than in the context of finding the desirable trajectory of motion. Nevertheless one cannot argue that the real problem of concern is finding the location and/or trajectory of motion which provides a desired value of some "goodness" measure (e.g. the value of some cost-function).

Generalization of the NHC decision-making process. Actually, the decision-making process in NHC is an incompletely observed Markov process. The space is being disctetized, the centers of tiles in the tesselation are connected and form a graph, cost of each edge of the graph is assigned. This allows for using an approach to discretization of the control system which is different from the existing approaches of systems "digitalization". We do not discretize the time by introducing a "sampling period". The system is being considered to move in a discretized state-space from one node on the graph to another one paying the cost of the edge upon its traversal.

Method of Nested Dynamic Programming (NDP) states that the optimum control should be found by joint 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.

This will obviously lead to a very fuzzy solution from the view of the lowest level of system ("actuator"). However, this allows for an advantage: a large part of the world will be excluded later from consideration at the lower levels.

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

The optimum solution for the upper level, is considered at the lower level as the stripe (zone, envelope of search) for further independent decision-making. However, due to the sensor information which appears at a given level during the performance, the optimum solution of the lower level may require to seek beyound the stripe of independent decision making. Rule 3 is to be applied in this case.

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.

Theorem 3. Given a nested world representation

$$S_1 \supset S_2 \supset S_3 \supset ... \supset S_i \qquad (11)$$

and a set of cost-functionals for these representations, based upon common policy of decision-making, the set of decisions will constitute a nested hierarchy

$$D_p\ (S_1) \supset D_p\ (S_2) \supset D_p\ (S_3) \supset ... \supset D_p\ (S_i)\ . \qquad (12)$$

A family of MHC-algorithms. The multiresolutional hierarchy of maps $\{M_i\}$,i=1,2,... obtained from the subsystem of sensors, and organized within the knowledge base, together with the initial location and goal assignment, are the inputs for control system. Actually, this is the multiresolutional hierarchy that should be checked on consistency. Another set of maps $\{m_i\}$,i=1,2,... is being generated in the process of interaction between the subsystems of knowledge representation and controller: each map of this new set of maps is a subset of the "larger picture" selected based on a premise that the maps $\{M_i\}$ are too large and working with them will require too much time and computing power. So, the subset is determined which is substantially smaller than the initial map, and yet it is possible to expect that the rest of the main map can be sacrificed with no harm to the problem solving process. Each of the maps-subsets can be evaluated by the probability of being sufficient for finding the optimum solution.The smaller the subset is the lower is the value of probability of being sufficient. This factor should be taken in account in the design.

The single control process is being produced using the operator of *search in the state space*, or MS^3-search (multiresolutional modified dynamic programming). All possible motions of the system are presented in the state space via measures for costs that can be computed for moving from one point of the state space to another. The space is discretized and the cost is computed for a motion from one discrete to the adjacent disretes. Then a minimum-cost trajectory can be searched for.

The process proceeds as follows. Firstly, a subset $m_n \in M_n$ is heuristically obtained from the initial low resolution map at the n-th level by *focusing attention* at the part of the map in which the process is expected to be executed. For the generality we assume that this initial subset is obtained by applying the search algorithm $S^3(J_n,M_n)=m_n$ to the initial complete map M_n under requirement to minimize the cost-functional J_n. An envelope is built which is a subset of the search space, includes the solution, and a vicinity of this solution. The technique can be applied of building a "convex hull" (see [39]) around the minimum path with the zone of uncertainty attached to the trajectory. This envelope (convex hull) will be called later in this paper $CH(m_n)=M_{n-1}$.

The result of CH operation is considered the map M_{n-1} for the subsequent search at the next (n-i)-th level of resolution.The process of interaction is developing furtherly as follows [26]:

$$
\begin{array}{lccccccc}
\text{line1:} & (m_1 \overset{fa}{\subset} M_1) & \overset{g}{\supset} & (m_2 \overset{fa}{\subset} M_2) & \overset{g}{\supset}\ldots\overset{g}{\supset} & (m_{n-1} \overset{fa}{\subset} M_{n-1}) & \overset{g}{\supset} & (m_n \overset{fa}{\subset} M_n) \\
 & \supset & & \supset & & \supset & & \supset \\
 & \downarrow & & & & \downarrow & & \downarrow \\
\text{line2:} & P/CA(J_1,M_1)=m_2 & \ldots & & \ldots & P/CA(J_{n-1},M_{n-1})=m_{n-1} & & P/CA(J_n,M_n)=m_n \\
 & \downarrow & & & & \downarrow & & \downarrow \\
\text{line 3:} & \text{output} & \ldots & & & CH(m_{n-1})=M_{n-2} & & CH(m_n)=M_{n-1} \\
 & \downarrow & & & & \downarrow & & \downarrow \\
 & \text{to controller} & \ldots & & & \text{to (n-2)-th} & & \text{to (n-1)-th} \\
 & & & & & & \text{levels} &
\end{array}
$$

Line 1 shows two multiresolutional hierarchies: one of them by generalization (of knowledge maps {Mi}) and another by focus of attention (of knowledge maps $\{m_i\}$). All signs $\supset$ are related to the relation of "focus of attention, while all signs $\subset$ are related to the relation of "generalization". Hierarchy of sets $\{m_i\}$ is obtained from the hierarchy of sets by applying planning/control algorithm (P/CA) per level (line 2). In order to compute a set of {Mi}, the results of applying P/CA-algorithm per level, are enhanced up to the meaningful consistent map partition [13]. So this system makes consecutive refinement in a closed loop fashioh, level-to-level and top-down. An example of algorithm is given here which has been applied in a number of intelligent controllers (MS^3-algorithm).

Multiresolutional State Multiresolutional Consecutive Refinement: Search in the Space (MS^3-algorithm). Search by scanning the string of available alternatives ("browsing"), and selection when the desirable property is met, is one of the straightforward algorithms of *combination generation.*

On the other hand, any combinatorial algorithm is an operator of generating *solution alternatives* for a decision-making process. 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 is called *policy of decision-making,* or policy of control). This search was actually executed as a part of research reported in [32-38].

The whole process is illustrated in Figure 12. Our experience with discretizing the space shows that the most beneficial results can be achieved by building a random graph (grid). The density of this grid can be introduced as the ratio

$$\rho = {}^{N}/_{S}$$

where N is the number of points at a level, and V is the total volume of the state space (under consideration).

The points are put in a uniform manner and the idea of the volume of the vicinity can be introduced which is the value inversely proportional to the density

$$\sigma = {}^{1}/_{\rho} = {}^{S}/_{N}.$$

This discretization can be characterized by the value of average distance between the two "adjacent nodes" of the graph. The idea of NHC-algorithms is illustrated in Figure 12. At the highest level the system is represented with low resolution (coarse granularity). Nodes of the graph representing the system are shown without edges which connect them. After the solution is found, an envelope is determined which contains the solution and a definite vicinity of it.One can expect that the width of the envelope should not exceed the width of the unit of the grid at this particular level[4] .

Example of an Algorithm.

I. Establish the *state space* in which the whole procedure will be executed.

1.1 Name and list the inputs, and their operating intervals

1.2 Name and list the outputs, and their operating intervals

1.3 Formulate all mappings known for the system to be analyzed: including its differential, algebraic, and logical statements.

[4] Determining the envelope is described in the example.

1.4 Select the cost-functional in which all costs will be computed.

1.5 Declare the data structures in which the states will be analyzed.

Comment: in the test example of applying this algorithm, the following structures were created:

(a) a node adjacency storage for the graph representation.

(b) a cost record which holds data indicating the node number, its cumulative cost from the source node.

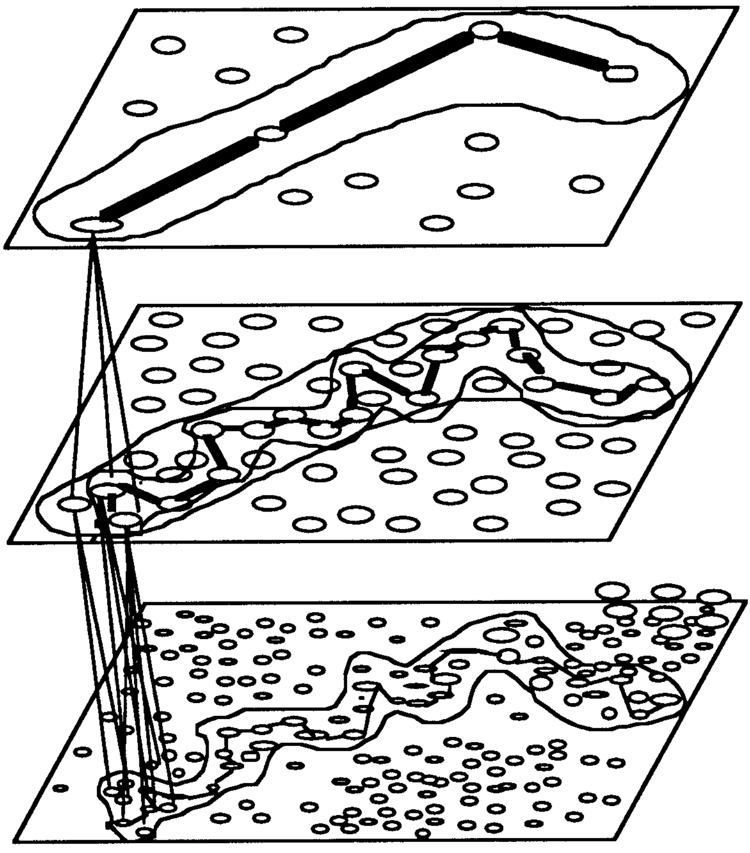


Figure 12. Illustration to the MS3 -algorithm

(c) a structure which contains only temporary nodes, its successor and predecessor (list OPEN).

(d) results of least-cost path algorithm which contains a node number, a corresponding cost, its successor and predecessor (list CLOSED).

(e) list of the paths at different resolution levels (PDR).

(f) list of the vicinities for each node stored in the list CLOSED.

II. Execute the NHC Procedure.

2.1 Generate random nodes at the required density within the zone under consideration (and store them).

Comments:

(a) Density is estimated by the number of nodes per unit of the area.

(b) The initial area of computation is determined by the full intervals of all inputs and all outputs.

(c) The second, third, (and so on until the final) zones in which the random nodes are generated are equal to the "vicinity" of the node (step 2.2).

2.2 Generate the vicinity for the first node.

Comments:

(a) Vicinity is a part of the state-space adjacent to the current point in which all random nodes previously generated are considered to be connected to the current point; sometimes, the vicinity is bounded by an ellipse.

2.3 Store the condition of vicinity in the form of an equation.

2.4 Check all nodes of the random graph and extract from them a subset which is located within the vicinity.

Comment: Only the subset located within the vicinity of the current node is considered the"successor" of the current node.

2.5 For the set of successors, do the following:

(a) compute the cost of motion from the current node to the successor.

(b) store all of the successors with their costs in the list OPEN.

(c) remove from the list OPEN the partial path (PP) node for which cost is minimum; resolve ties arbitrarily.

2.6 IF PP is the goal node, GOTO 2.9 with the solution.

2.7 ELSE Put PP on the list CLOSED.

2.8 GOTO 2.2., considering the point from 2.7 the initial point

2.9 Put the solution on the PDR list.

Comment:

(a) Optional: draw the path at a particular resolution

(b) Clearly, for the best path which has been found at a particular resolution, we have the string of their nodes and the vicinity constraints which have been remembered.

2.10 Put n random points in each vicinity of all points of the path under consideration; in the areas where vicinities overlap, merge all couples of points which have distance between them smaller than the distance at the level of resolution under consideration.

Comments:

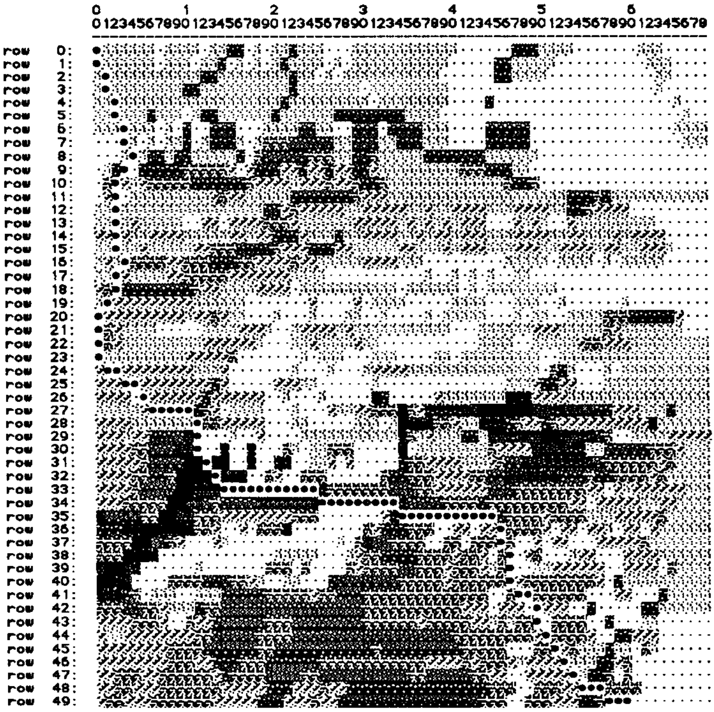
(a) The number n of points to be put in the vicinity depends on the resolution level.

(b) Merger of the points which are located too close to each other can be done in different ways: by finding a midpoint, etc.

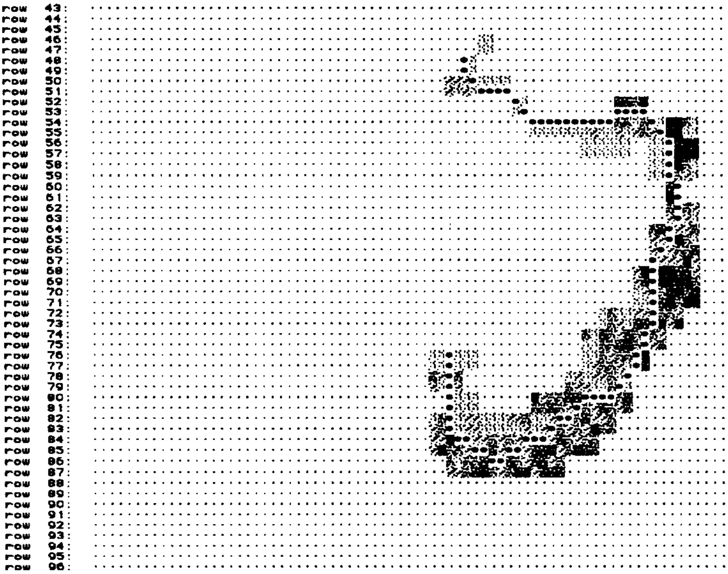
2.11 IF the density in the new vicinity is higher than is required by conditions of accuracy, THEN exit with FINAL SOLUTION

2.12 ELSE GOTO 2.2 considering the new vicinity size.

In Figure 13 the results of the broad low resolution search, and the narrow high resolution search (in a sripe) are shown for a case of planning a path in a terrain with expert assignment of costs [32].



. a



.b

Figure 12.The results of low-resolution search (a), and the refinement in the vicinity with the subsequent higher-resolution search (b)

Conclusions.

1. Theory of Nested Hierarchical Control is a development of theories of multiresolutional (multiscale, nested, multigranular) image representation, and multiresolutional (multiscale, mul;tiple time-scale) signal representation into the domain of Control Theory. NHC-theory allows for solving numerous problems of control in systems with incomplete, and/or inadequate information, large systems, autonomous control system, particularly in the area of intelligent machines.

2. NHC-theory treats design and control processes as a design-control continuum. Thus, planning is becoming a stage within this continuum, a case of off-line finding the open-loop control sequence. The open-loop/closed-loop control structure is becoming a module in a multiresolutional hierarchy of NHC-controller.

3. Algorithms of Multiresolutional Consecutive Refinement (or S^3-search) has been developed which allow for time-efficient computation of control sequences at all levels; it has some serious advantages in comparison witgh algorithms of dynamic programming.

References.

[1]. M. D. Mesarovic, "On Self Organizational Systems", Eds.: M. Yovits, G. T. Jacobi, G. D. Goldstein, Self-Organizing Systems-1962, Spartan Books, Washington, DC 1962

[2]. G. Pask, "Interaction Between a Group of Subjects and an Adaptive Automation to Produce a Self-Organizing System for Decision Making", ibid.

[3]. M. Mesarovic, D. Macko, Y. Takahara, Hierarchical Multilevel Systems Theory, Academic Press, New York, 1970

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

[5]. J.S. Albus, "Mechanisms of Planning and Problem Solving in the Brain"," *Mathematical Biosciences"*, v. 45, 1979

[6]. G. Giralt, R. Sobek, R. Chatila, "A Multilevel Planning and Navigation System For a Mobile Robot: A First Approach to Hilare", Proc. of IJCAI-79, Vol. 1, Tokyo, 1979

[7]. J. S. Albus, Brains, Behavior, and Robotics, BYTE Books/McGraw-Hill, Peterborough, NH 1981

[8]. 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

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

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

[11]. 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

[12]. 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

[13]. 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

[14]. J.S. Albus, C.R. McLean, A.J. Barbera, M.L. Fitzgerald, "Hierarchical Control for Robots, and Teleoperators", Proc. of the Workshop on Intelligent Control, Troy, NY, 1985

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

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

[17]. 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, 1986

[18]. 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

[19]. A. Meystel, Primer on Autonomous Mobility, Drexel University, Philadelphia, PA, 1986

[20]. 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

[21]. A. Meystel, "Nested hierarchical controller for intelligent mobile autonomous system," Proc. Int'l Congress on Intelligent Autonomous Systems, Amsterdam, The Netherlands, 1986

[22]. A. Meystel, "Knowledge-based Controller for Intelligent Mobile Robots", In Artificial Intelligence and Man-Machine Systems, Ed. by H. Winter, Series "Lecture Notes in Control and Information Systems",v. 80, Springer Verlsg, Berlin, 1986

[23]. A. Meystel, "Nested Hierarchical Intelligent Module for Automatic Generation of Control Strategies", Languages for Sensor-based Control in Robotics, Ed. by U. Rembold, K. Hormann, Springer-Verlag, Berlin, 1987

[24]. A. Waxman, et. al., A Visual Navigation System for Autonomous Land Vehicles, *IEEE J. of Robotics & Automation*, Vol. 3, No. 2, 1987

[25]. A. Meystel, "Planning in a hierarchical nested autonomous control system," Proc. SPIE Symp., Vol. 727, Cambridge, MA, Oct 30-31, 1987

[26]. A. Meystel, "Theoretical Foundations of Planning and Navigation for Autonomous Mobile Systems", *International Journal of Intelligent Systems*, Vol.2, No.2, 1987

[27]. A. Meystel, et al.,"Multiresolutional pyramidal knowledge representation and algorithmic basis of IMAS-2," Proc. of Mobile Robots, SPIE, Vol. 851, 1987

[28]. A. Meystel, "Nested hierarchical controller with partial autonomy," Proc. Workshop on Space Telerobotics, V1, NASA, JP, Pasadena, CA, July 1, 1987

[29]. C. Isik, A. Meystel, "Pilot level of a hierarchical controller for an unmanned mobile robot," *IEEE J. of Robotics & Automation*, Vol. 4, No. 3, 1988

[30]. A. Meystel, "Intelligent control of robots," *J. of Robotic Systems*, Vol. 5, No. 4, 1988
[31]. A. Meystel, "Mobile robots, autonomous," in: Int'l Encyclopedia on Robotics; editor, R. Dorf, Wiley Publishing Company, 1988
[32]. G. Grevera, A. Meystel, "Searching for a path through Pasadena", Proc. of the IEEE Symposium on Intelligent Control, Arlington, VA,1989
[33]. A. Meystel, "Multiresolutional Control System", Proc. of the IEEE Symposium on Intelligent Control, Arlington, VA 1989
[34]. R. Bhatt, D. Gaw, A. Meystel, "Learning in a Multiresolutional Conceptual Framework", Proc. of the IEEE Int'l Symposium on Intelligent Control 1988, IEEE Comp. Soc. Press, 1989
[35]. A. Meystel, Knowledge-Based Nested Hierarchical Control", in Knowledge-based Systems for Intelligent Automation, Vol. 2, Ed. by G. Saridis, JAI Press, Greenwich, CT 1990
[36]. A. Meystel, S. Uzzaman, G. Landa, S. Wahi, R. Navathe, B. Cleveland, "State Space Search For An Optimal Trajectory", Proc. of the IEEE Symposium on Intelligent Control, Vol. II, Philadelphia, PA, 1990
[37]. B. Cleveland, A. Meystel, "Predictive Planning+Fuzzy Feedback Compensation=Intelligent Control", Proc. of the IEEE Symposium on Intelligent Control, Vol. II, Philadelphia, PA, 1990
[38]. G. Grevera, A. Meystel, "Searching in a Multidimensional Space", Proceedings of the IEEE Symposium on Intelligent Control, Vol. I, Philadelphia, PA,1990
[39]. A. Meystel, Autonomous Mobile Robots : Vehicles with Cognitive Control, World Scientific, 600 pages, 1991
[40]. A. Meystel, "Multiresolutional Recursive Design Operator for Intelligent Machines", Proc. of the IEEE Int'l Symp. on Intelligent Control, Arlington, VA 1991
[41]. A. Meystel, "Multiresolutional Feedforward-Feedback Loops", Proc. of the IEEE Int'l Symp. on Intelligent Control, Arlington, VA 1991
[42]. A. Meystel, "Coordination in a Hierarchical Multiactuator Controller", Proc. of the NASA Conference on Space Telerobotics, Pasadena, CA 1989
[43]. J. Albus, "A Reference Model Architecture for Intelligent Systems Design", in An Introduction to Intelligent and Autonomous Control, eds. P. J. Antsaklis, and K. M. Passino, Kluwer, 1992
[44]. B. P. Zeigler, S. Chi, "Model-Based Architecture Concepts for Autonomous Systems Design and Siomulation", in An Introduction to Intelligent and Autonomous Control, eds. P. J. Antsaklis, and K. M. Passino, Kluwer, 1992
[45]. K. M. Passino, P. J. Antsaklis, "Modeling and Analysis of Artificially Intelligent Planning Systems", in An Introduction to Intelligent and Autonomous Control, eds. P. J. Antsaklis, and K. M. Passino, Kluwer, 1992
[46]. P.K.S. Wang, "A Method for Approximationd Dynamical Processes by Finite State Systems", *Intern J. Control*, Vol.8, No.3, 1968
[47]. Y. Bar-Shalom, "Stochastic Dynamic Programming: Caution and Probing", *IEEE Transactioms on Automatic Control*, Vol. AC-26, NO. 5, Oct. 1981
[48]. S. Gnesi, U. Montanari, A. Martelli, "Dynamic Programming as Graph Searching: An Algebraic Approach", *J. of the ACM*, Vol. 28, No.4, Oct. 1981
[49]. J. Pearl, Heuristics, Addisson-Wesley, Reading, MA 1984

[50]. J.-P. Forestier, P. Varaiya, "Multilevel Control of Large Markov Chains", *IEEE Transactions on Automatic Control*, Vol. AC-23, No.2, April 1978
[51]. W. Hackbush, Multi-grid Methods and Applications, Springer-Verlag, Berlin, 1980
[52]. A. Brandt, Guide to Multigrid Development, in Multigrid Methods, Lecture Notes in Mathematics 960, eds. W. Hackbush and U. Trottenberg, Springer-Verlag, Berlin, 1982
[53]. A. Brandt, D. Ron, D. Amit, "Multi-Level Approaches to Discrete States and Stochastic Problems", in Multigrid Methods II, Lecture Notes in Mathematics 1228, eds. W. Hackbush and U. Trottenberg, Springer-Verlag, Berlin, 1986
[54]. M. Minsky, "A Framework for Representing Knowledge", in P. Winston (ed.), The Psychology of Computer Vision, McGraw-Hill, 1975
[55]. B. Mandelbrot, The Fractal Geometry of Nature, W. H. Freeman and Co., 1977, 1983
[56]. H. Samet, Applications of Spatial Data Structures, Addison-Wesley, 1990
[57]. A. Rosenfeld (ed.), Multiresolutional Image Processing and Analysis, Springer-Verlag, New York, 1984
[58]. A. Meystel, "On the Phenomenon of High Redundancy in Robotic Perception", in Highly Redundant Sensing in Robotic Systems, ed. by J. Tou, J. Balchen, Springer-Verlag, Berlin, 1990
[59]. S. Tanimoto, "Paradigms for Pyramid Machine Algorithms", in Pyramidal Systems for Computer Vision, eds. V. Cantoni, S. Levialdi, Springer-Verlag, Berlin, 1986
[60]. L. Uhr, Algorithm-structures Computer Arrays and Networks: Architectures and Processes for Images, Precepts, Models, Information, Academic Press, New York, 1984
[61]. M. Coderch, A. Willsky, S. Sastry, D. Castanon, "Hierarchical Aggregation of Linear Systems with Multiple Time Scales", *IEEE Transactions on Automatic Control*, Vol. AC-28, No. 11, November 1983
[62]. S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, July 1989
[63]. M. Basseville, A. Benveniste, K. C. Chou, A. S. Willsky, "Multiscale Statistical Signal Processing: Stochastic Processes Indexed by Trees", MTNS 89, June 19-23, Amsterdam, 1989
[64]. A. Benveniste, R. Nikoukhah, A. S. Willsky, "Multiscale System Theory", internal publication No. 518, IRISA, INRIA, February 1990
[65]. U. Ozguner, "Near-optimal Control of Composite Systems: the Multi Time-Scale Approach", *IEEE Transactions on Automatic Control*, Vol. AC-24, No.4, April 1979
[66]. B. Litkouhi, H. Khalil, "Multirate and Composite Control of Two-Time Scale Discrete-Time Systems", *IEEE Transactions on Automatic Control*, Vol. AC-30, No. 7, July 1985
[67]. W. Findeisen, et al., Control and Coordination in Hierarchical Systems, Wiley, New York, 1980