N90-29820

DESIGN OF A STRUCTURAL AND FUNCTIONAL HIERARCHY FOR PLANNING AND CONTROL OF TELEROBOTIC SYSTEMS

Levent Acar
Dept. of Electrical Engineering
University of Missouri-Rolla
112 Electrical Engineering
Rolla, MO 65401

Ümit Özgüner
Dept. of Electrical Engineering
The Ohio State University
2015 Neil Avenue
Columbus, OH 43210

Abstract

Hierarchical structures offer numerous advantages over conventional structures for the control of telerobotic systems. A hierarchically organized system can be controlled via undetailed task assignments and can easily adapt to changing circumstances. The distributed and modular structure of these systems also enables fast response needed in most telerobotic applications. On the other hand, most of the hierarchical structures proposed in the literature are based on functional properties of a system. These structures work best for a few given functions of a large class of systems. In telerobotic applications, all functions of a single system needed to be explored. This approach requires a hierarchical organization based on physical properties of a system. In this paper, such a hierarchical organization is introduced. The decomposition, organization and the control of the hierarchical structure are considered, and a system with two robot arms and a camera is presented as an example.

1. Introduction

In most telerobotic applications, the need to express undetailed tasks and to expect the robotic system to plan and reason about its environment is essential. Hierarchical structures provide systematical methods for planning detailed task assignments and deriving fine-motion control strategies.

There are many other reasons to create hierarchical organizations for telerobotics systems. Most importantly, these organizations help to simplify the controller design by allowing designs on smaller portions of a large, complex system. They also offer distributed computation capabilities and enable local reasoning and planning which are necessary for fast response and error recovery.

Hierarchical structures are based on different representations of a system in Control and in Artificial Intelligence (AI) Theory. In Control, a system is usually defined by its dynamic equations. As a result, most hierarchical distributions in Control exploit mathematical properties in the description of the system. These decompositions range from determining coupling parameters inside system dynamics to input-output correlations of a system, [1]–[7]. On the other hand in AI, organizations are based on functional behavior and accomplishable goals, [8, 9]. These goals are usually divided into simpler subgoals which form the levels of the hierarchy. Similar hierarchies were also utilized for computational and design purposes, [10]-[12].

To plan control strategies and execute them, the two type of representatives should be unified. In the literature, this unification was addressed in two different approaches. The first approach was a fixed level hierarchy advocated by Albus et al. [13]-[16] and others [17]. Albus used a five level hierarchy where the system is connected to the structure at the last level. This approach was successfully implemented for the control of a robot arm, but it is not general enough to include more complicated systems with numerous functional behaviors. The second method was by Saridis et al. [18]-[20] and it used a three level classification hierarchy with organizational, coordination and executional levels. Similar to the first approach, the system was connected to the hierarchy at the last level. The coordination level could be viewed as a translator between the functional and the mathematical representations describing the system. This hierarchy is more general, because it was explained with vague descriptions, such as the intelligence decreases as the detail and granularity increases. Similar organizations under different names were also given in [21]-[23].

There are two major similarities between the two organizations. Both of them form tree structured hierarchies. However, as pointed out in [24], most of the hierarchical structures have more complicated connections, and levels of hierarchy may not be obvious. Both of these organizations work well for a wide range of systems provided only a few functional behaviors are expected. On the other hand, most telerobotic applications require a system to work well for a variety of functions.

In this paper, a structural and functional hierarchy is proposed to overcome these problems and to obtain an alternative structure. The hierarchy is first formed by considering the physical properties of a system. Then, functions are associated with this organization. Finally, a uniform control flow is described for the hierarchy. A two robot and a camera system is presented as an example to demonstrate the details and the applicability of the structure.

2. Hierarchical Structure and Control

In this section, we introduce the general form of a hierarchical organization for the control of telerobotic systems. The structure consists of functional abstractions associated with a physical decomposition of the system.

We start the decomposition by separating the system into an initial set of components. These components don't have to be disjoint or complete, but as we will observe later, the detail of the hierarchy depends on this initial choice.

The second step will be to obtain the largest set of disjoint components from the initial set. This set of disjoint components will form the bottom level of the hierarchy and the whole system will form the top level. In between these two levels, there are numerous components connected as a directed graph with no structural loop. These mid-structure components must be connected so that there is always a commom portion between a node and any of its

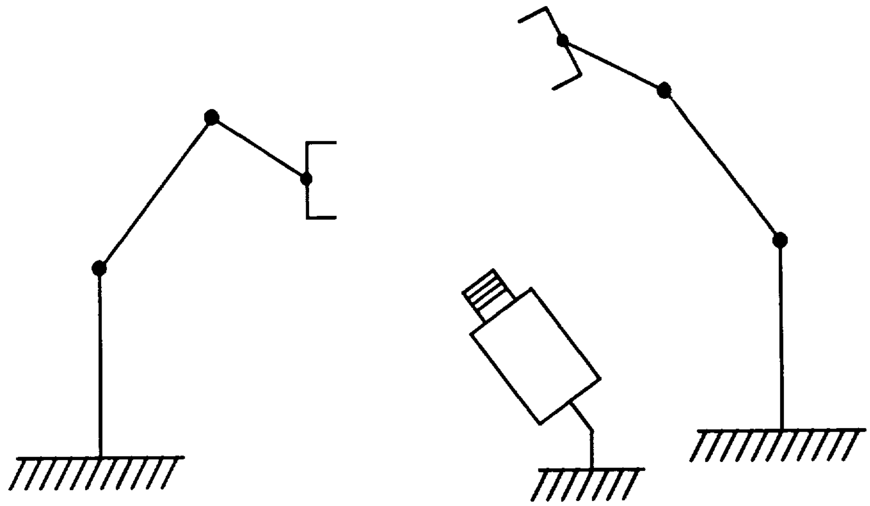

Figure 1: A two robot and one camera system.

subnodes[1], and the collection of all the subnodes of a node contains at least the node itself. These requirements produce a very interesting hierarchical structure where the whole system is represented almost completely at different levels of detail within the hierarchical organization. The top level is the least and the bottom level is the most detailed level. Although, the levels in the middle are not necessarily well defined, selective collections of components from the mid-structure would describe the system at different details.

To see an example of a hierarchical organization, we consider a system with two robot arms and one camera as in Figure 1. One possible hierarchical organization is given in Figure 2, where the dashed boxes represent the initial set of components.

After the formation of a hierarchical organization, functions related to the components are assigned and the flow of the control process is described. To represent the functions and the control process, we introduce six primitives. The primitives exist at every node of the hierarchy and related to each other through the control process. These primitives are:

Goals are assertions representing the end result to be obtained.

Tasks are elementary job descriptions.

Procedures are methods of accomplishing tasks. Procedures are separated into two groups:

1. *Current Procedures* which are locally applicable procedures, and
2. *Subprocedures* which are applicable only by subnodes.

Measurements are the available information from sensors. The measurements are also separated into two groups:

1. *Current Measurements* which are the measurements available locally, and
2. *Other Measurements* which are the measurements of other nodes.

[1]subnodes are sometimes called children nodes

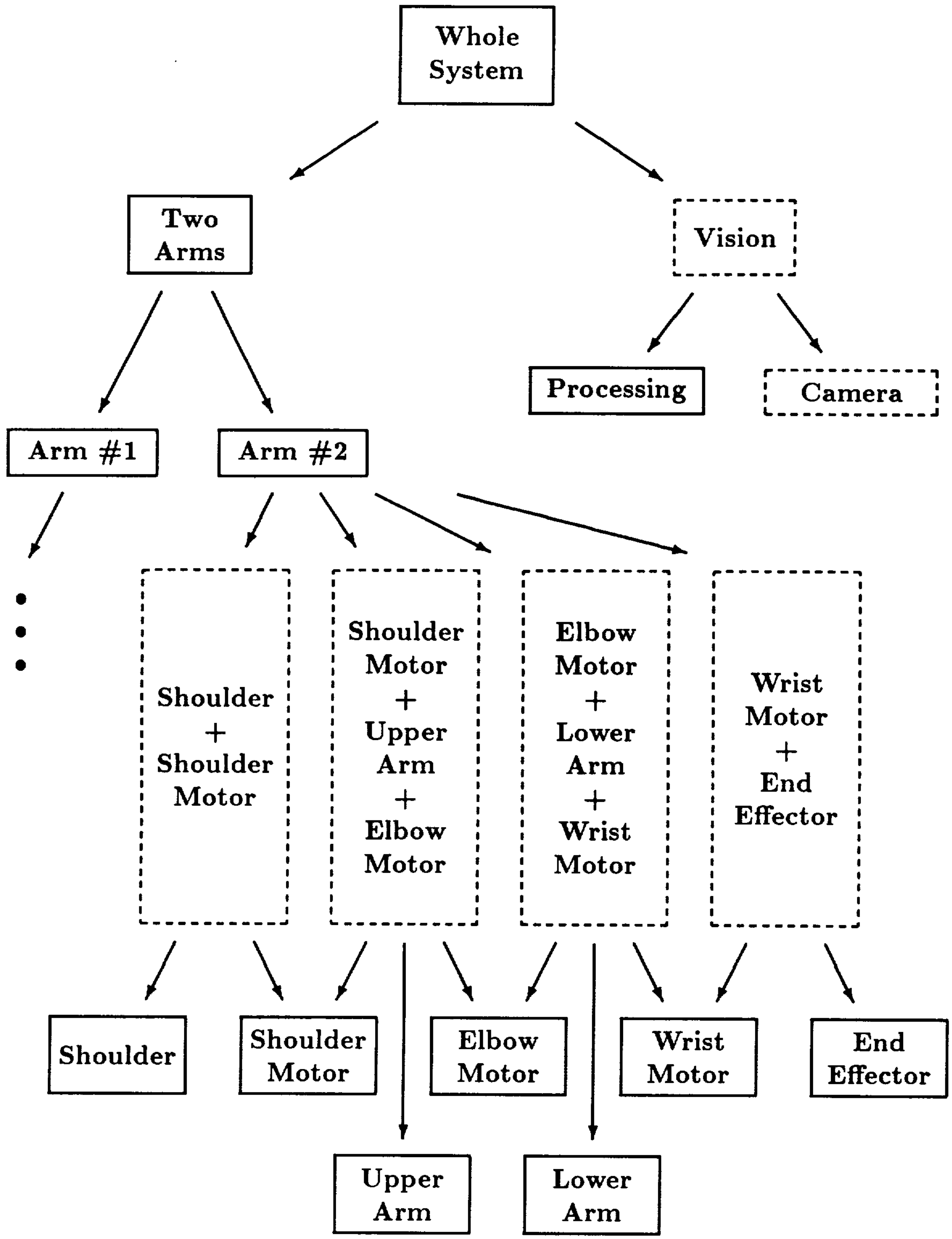

Figure 2: One possible hierarchical structure of the two robot and one camera system.

Constraints are task restrictions or exemptions which are formed by:

1. *System Dependent Constraints* which depend solely on the system and do not change with the environment or the goals, such as angular limitations of a robot arm due to its construction, and
2. *System Independent Constraints* which does not depend on the system, but depend on the environment or on the assigned goals, such as angular limitations of a robot arm due to an obstacle.

Resources are task restrictions or exemptions which are related to the use of procedures. Similar to measurements, resources are also separated into two groups:

1. *Current Resources* and
2. *Other Resources.*

We represent functions of the components by collections of tasks. At each node, a list of tasks with associated procedures, constraints and measurements forms the knowledge base of the node. Procedures associated with the same task provide different ways of accomplishing the task. Utilizing a current procedure implies that the associated task can be accomplished at that node and there is no need to propogate the task further down in the hierarchy. On the other hand, utilizing a subprocedure implies goals have to be formed for the subnodes to accomplish the task. This information provided by the subprocedures is important for timely propogation of tasks in the hierarchy.

In the proposed organization, a procedure may accomplish more than one task and a set of procedures may accomplish a single task. If we need to order tasks and procedures sequentially, additional constraints are included to synchronize the execution.

Procedures also use measurements and consume or produce resources. We include the information about some of the measurements available at other nodes as part of their knowledge bases. This information helps to utilize other measurements directly without the use of usual backward and forward search methods. The knowledge about the resources of other nodes is included for failure handling purposes only, [25], and it is not used for control.

Constraints are also ranked among themselves. We assign a *stiffness constant* to each constraint according to its importance. For example, a constraint on the allowable grasping tension for a robot end-effector can be relaxed if the robot is holding a steel pipe, but it has to be observed strictly if the robot is holding a fragile vase.

We also assign another type of constant for resources and measurements to represent the cost of using them. These costs affect the choice of procedures which use measurements and resources.

The six primitives are related to each other in a unique way by the control process. Every node has the identical process as shown in Figure 3. When a goal with constraints and resources is received by a node, it is first decomposed into a number of tasks such that the accomplishment of the tasks implies the accomplishment of the goal. Since many procedures may be assigned to a task, one set is selected according to a cost criterion. The cost may be energy, entropy or currency as long as an optimum exists. Among the selected procedures, some may also be subprocedures. In the last stage, these subprocedures are used to form goals for the subnodes. The stiffness constants of the corresponding tasks are also propogated with the goals.

Most of the knowledge is embedded into the system before the system starts execution. The dashed boxes in Figure 3 show the portion of the process which runs in real-time after goal assignments.

The connections of the hierarchy are initially determined before the task assignments. These connections form a static structure. When the system starts running, nodes can be temporarily connected with each other to exchange measurements. This exchange introduces a dynamic structure which may create loops in the organization. However, these loops don't cause any cyclic behavior, since only measurements can be exchanged via the new connections. Without this dynamic structure, the controlling ability of the organization would have been severely limited.

3. Functional Assignments

Next, we will briefly discuss task assignments and goal propogation process for a two robot arm and one camera system introduced earlier. We will only consider a few of the applicable tasks to give a general view of the functional behavior of the structure.

We assume that the system is asked to carry an object from one end of a table to the other end, where both arms have to be utilized. At the top level, the goal *Carry* is matched with the task *Carry*. We assume that there are three procedures associated with task *Carry*: *Carry_alone*, *Carry_separately* and *Carry_together*. Procedure *Carry_alone* utilizes only one arm. Procedure *Carry_separately* picks the object with one arm and puts it on the table between the two arms for the other arm to carry it further away. *Carry_together* transfer the object in the air without putting it on the table. With the help of the *Vision* node which determines distances between locations, the top node decides to use the *Carry_together* procedure. Since all these procedures are subprocedures, a goal for the *Two_Arms* node is formed and decomposed into the following tasks:

GOAL $\longrightarrow$	TASK DECOMPOSITIONS
Carry_together $\longrightarrow$	*Reach* ?stiffness Arm#1 Location_Object
	Lift ?stiffness Arm#1 Location_Object
	Reach ?stiffness Arm#1 Midlocation_1
	Reach ?stiffness Arm#2 Midlocation_2
	Transfer ?stiffness Arm#1 Arm#2
	Reach ?stiffness Arm#1 Location_final
	Reach ?stiffness Arm#2 Destination
	Putdown ?stiffness Arm#2 Object Destination

Constraints are also added to these tasks to preserve the synchronization and the sequential order. The status of the constraints are obtained from the *Vision* node. In the next stage,

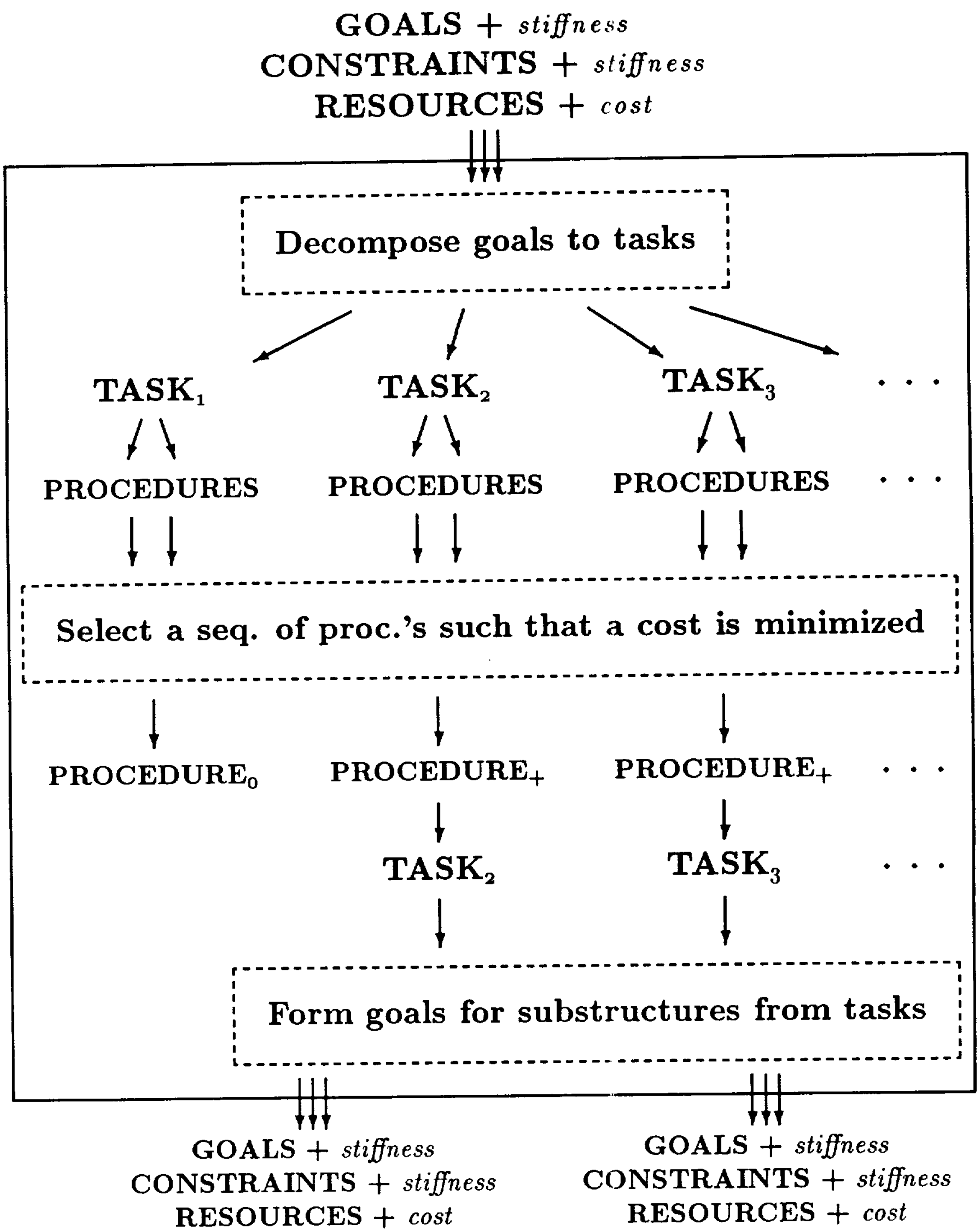

Figure 3: Control flow diagram for each node.

goals for the *Arm#1* and *Arm#2* nodes are formed from these tasks. After a similar decomposition, elementary tasks such as *Move_Coarse*, *Move_Detailed*, *Orient*, *Grasp* and *Ungrasp* are formed. This process continues down to the bottom level nodes where the tasks and procedures are more elementary. Changing joint angles, opening or closing the end-effector and the input voltage profiles are among the tasks of these nodes.

4. Conclusions

In this paper, we considered a hierarchical organization primarily based on the physical properties of a system which is most suited for telerobotics applications.

We formed the hierarchy as a directed graph with no structural loop from the components of the system. We arranged the components so that the system is described with least detail at the top level and with most detail at the bottom level. In between these two levels, the system can be described with any desired detail depending on an initial choice of components. We also did not limit the type of the hierarchy[2]. We deliberately allowed representational redundancies in the middle levels, but we forced the top and the bottom levels to be represented without redundancies. This restriction is to enforce a consistent control at the bottom level.

Then, we assigned knowledge and functionality to the nodes of the hierarchy by using six primitives: Goals, Tasks, Procedures, Measurements, Constants and Resources. These primitives contain the knowledge about the capabilities of the components and the information about their behavior.

Finally, we described the control process as decomposition of goals into tasks, selection of procedures for these tasks and formation of new goals for the subnodes. This process starts at the top level and propogates down uniformly. As control decisions are made at the nodes of the hierarchy, we allowed new connections among nodes for data exchange. These connections save time and make efficient use of all possible control strategies.

To observe the formation of the proposed hierarchy and the propogation of the control process, we considered a system with two robot arms and a camera.

References

[1] M. Jamshidi. *Large-Scale Systems.* North-Holland, Amsterdam, Netherlands, 1983.

[2] F. Harary, R. Z. Norman, and D. Cartwright. *Structural Models: An Introduction to the Theory of Directed Graphs.* John Wiley & Sons, Inc., New York, New York, 1965.

[3] D. P. Looze and N. R. Sandell, Jr. Hierarchical control of weakly-coupled systems. *Automatica*, 18(4):467–471, July 1982. *Brief Paper.*

[2]In the literature, most hierarchies are limited to the tree structured hierarchies.

[4] A. Vannelli. *Solution Techniques for 0-1 Indefinite Quadratic Problems with Applications to Decomposition.* PhD thesis, Department of Electrical Engineering, University of Waterloo, Waterloo, Canada, 1983.

[5] M. E. Sezer and D. D. Šiljak. Nested ε-decompositions and clustring of complex systems. *Automatica*, 22(3):321–331, May 1986.

[6] A. H. Nayfeh. *Perturbation Methods.* John Wiley & Sons, Inc., New York, New York, 1973.

[7] V. I. Utkin, S. V. Drakunov, D. E. Izosimov, A. G. Lukyanov, and V. A. Utkin. A hierarchical principle of the control system decomposition based on motion separation. In *Preprints of the 9th World Congress of the International Federation of Automatic Control: Volume V*, pages 134–139, Budapest, Hungary, July 1984.

[8] P. H. Winston. *Artificial Intelligence.* Addison-Wesley Publishing Company, Inc, Reading, Massachusetts, 2nd edition, July 1984.

[9] E. Rich. *Artificial Intelligence.* McGraw-Hill Book Company, New York, New York, 1983.

[10] D. L. Zeltzer. *Representation and Control of Three Dimensional Computer Animated Figures.* PhD thesis, Department of Computer and Information Science, The Ohio State University, Columbus, Ohio, August 1984.

[11] D. C. Brown and B. Chandrasekaran. Knowledge and control for a mechanical design expert system. *IEEE Computer Magazine*, 19(7):92–100, July 1986.

[12] D. C. Brown. *Expert Systems for Design Problem-Solving using Design Refinement with Plan Selection and Redesign.* PhD thesis, Department of Computer and Information Science, The Ohio State University, Columbus, Ohio, 1984.

[13] J. S. Albus, A. J. Barbera, and M. L. Fitzgerald. Hierarchical control for sensory interactive robots. In *Proceedings of the 11th International Symposium on Industrial Robots*, pages 497–505, Japan, October 1981.

[14] J. S. Albus, A. J. Barbera, and M. L. Fitzgerald. Programming a hierarchical robot control system. In *Proceedings of the 6th International Robot Technology*, pages 505–517, Paris, France, June 1982.

[15] J. S. Albus, C. R. McLean, A. J. Barbera, and M. L. Fitzgerald. Hierarchical control for robots and teleoperators. In *Proceedings of the IEEE Workshop on Intelligent Control*, pages 39–49, Troy, New York, August 1985.

[16] J. S. Albus, R. Lumia, and H. McCain. Hierarchical control of intelligent machines applied to space station telerobots. In *Proceedings of the Workshop on Space Telerobotics: Volume I*, pages 155–165, Pasadena, California, July 1987. Jet Propulsion Laboratory.

[17] J. Rasmussen. The role of hierarchical knowledge representation in decisionmaking and system management. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(2):234–243, March/April 1985.

[18] G. N. Saridis. Foundations of the theory of intelligence controls. In *Proceedings of the IEEE Workshop on Intelligent Control*, pages 23–28, Troy, New York, August 1985.

[19] G. N. Saridis and K. P. Valavanis. Software and hardware for intelligent robots. In *Proceedings of the Workshop on Space Telerobotics: Volume I*, pages 241–249, Pasadena, California, July 1987. Jet Propulsion Laboratory.

[20] K. P. Valavanis. *A Mathematical Formulation for the Analytical Design of Intelligent Machines.* PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, 1986.

[21] A. Meystel. Nested hierarchical controller with partial autonomy. In *Proceedings of the Workshop on Space Telerobotics: Volume I*, pages 251–270, Pasadena, California, July 1987. Jet Propulsion Laboratory.

[22] J. Jiang and R. Doraiswami. Performance monitoring in expert control systems. In *Preprints of the 10th World Congress on Automatic Control: Volume 6*, pages 303–307, Munich, Federal Republic of Germany, July 1987.

[23] W. J. Wolfe and S. D. Raney. Distributed intelligence for supervisory control. In *Proceedings of the Workshop on Space Telerobotics: Volume I*, pages 139–148, Pasadena, California, July 1987. Jet Propulsion Laboratory.

[24] H. A. Simon. *The Sciences of the Artificial.* The MIT Press, Cambridge, Massachusetts, 2nd edition, 1981.

[25] L. Acar and J. R. Josephson. Hierarchically distributed real-time replanning and plan execution with specialized expert systems. *(In preparation)*, 1989.

NASA GODDARD SPACE FLIGHT CENTER