

# The Industrial Robot

An International Quarterly Journal on Industrial Robot Technology

## One Man's View

Bernard M. Sallot, Executive Director, Robot Institute of America. 43

## Coming Events

3rd Industrial handling exhibition	45
1st Joint international robots conference	45
Performance evaluation of robots	45
Automation of interplant transport	45
Automation in warehousing	46

## FEATURE ARTICLES

Multi-grippers hot forge manipulator M. S. Konstantinov and Z. I. Zankov Higher Institute of Mechanical and Electrical Engineering, Bulgaria.	47
Technical and Economical consideration concerning industrial robots K. G. Lassi R. Kaufeldt AB, Sweden.	56
A practical visually interactive robot handling system. W. B. Heginbotham, C. J. Page and A. Pugh. University of Nottingham, England.	61
Control of intelligent robots and goal orientated languages G. Gini and M. Gini, Politecnico di Milano, Italy.	67

## The USA SCENE

5th I.S.I.R. Chicago, September 22nd—24th, 1975. 75

## The JAPANESE SCENE

Business program for 1975 77

## New Products & Developments

Television hits bottom	81
Electrolux industrial robots now available in U.K.	81
VFW—Fokker gantry type robot	82
British interest in robots is rising rapidly	82
A new Italian robot	83

## Industrial Robot

## ABSTRACTS

Informative abstracts of recently published articles, papers, reports etc from world-wide sources 85

Note for Contributors 94

Editor-in-Chief  
Professor W.B. Heginbotham, B.Sc. Tech.,  
M.Sc. Tech., Ph.D., F.I.Prod.E.,  
M.I.Mech.E., F.R.S.A.

Associate Editor  
Wendy A. Thornton, B.Sc

Art Editor  
David O'Leary L.S.I.A.

Subscription Manager  
Mrs Brenda Peritton

Publisher  
T.E.Brock, F.Inst.D., M.I.Inf.Sc.

The Industrial Robot is published quarterly by International Fluidics Services Ltd., Carlton, Bedford MK43 7JS England, (Tel: 0234 - 720479)  
The journal is available on Subscription at £22.00. (UK) US\$60.00 (Overseas) per volume

©Copyright 1975 International Fluidics Services Ltd.

The Industrial Robot is printed by The Cotswold Press, Eynsham, Oxford.

The Industrial Robot technical articles are indexed in Engineering Index, U.S.A.

# Control of intelligent robots and goal orientated languages

Giuseppino Gini, Maria Gini,  
Politecnico di Milano, Italy.

## I - INTRODUCTION

The purpose of this paper is to describe a robot planning system for controlling a robot. This system can conceive and execute plans enabling the robot to accomplish certain tasks such as moving boxes from one room to another in a simple but real environment.

Although these sorts of problems are commonly thought to demand a little degree of intelligence, they pose important conceptual problems and can require quite complex planning and execution strategies.

The exigence to provide a robot with autonomous deductive capability is related with the availability, on the computer, of a suitable deduction language. By such a language it is possible to project a program which enables the robot to plan its actions for achieving a goal in a specified world.

Giving the description of the initial state, the final state and the available laws, the program generates the 'plan' for the robot, i.e. finds a sequence of primitive actions which permit the transition from the initial to the final state. The sequence of physical actions which the robot must execute can correspond to this sequence.

In this paper a deductive system based on the use of goal oriented languages and, particularly, of the MICROPLANNER is described.

We have focused in particular on the problem of hierarchical planning: the plan is in terms of high-level actions, and the execution of any of these actions will involve the generation of a plan at a more detailed level, and so on down a hierarchy of details levels. A hierarchical plan generating system is, for example, a basis for learning new actions composed of a sequence of more basic actions.

Our system operates in a simple simulated environment. The program is written in MICROPLANNER and makes use of several of its features, like the pattern-directed procedures invocation, the automatic backtracking, the retrieval and storage of expressions in data base.

We begin to describe the meaning of intelligent behaviour of a robot and we examine the way in which this behaviour can be obtained.

In the 3rd Section we illustrate the robot plan formation and the world modeling features in relation with the proposed problem. In the 4th Section the design criteria for programming a robot which must operate in a world of rooms, doors and boxes are illustrated and the implementation is discussed.

In the last Section we present some considerations and we propose further developments for the robot problem solving.

In Appendix there are some examples realized by our system.

## II - ROBOT PROBLEM SOLVING AND GOAL ORIENTED LANGUAGES

By 'robot' we do not intend a machine that executes a preprogrammed sequence of actions; we consider instead a robot as a computer controlled mechanism that can interact with its world environment in an autonomous and reasonably intelligent way.

A robot therefore is a complex machine with sensor for collecting informations about its world, with deductive methods for solving problems and with movable parts for realizing them. The robotic is concerned with the study of vision systems, with the project of interfaces between the controlling computer and the motor and sensory hardware devices, and with the implementation of deductive systems.

Our interest is to illustrate and to discuss the construction of such a deductive system, i.e. a program able to perform intelligently a problem solution.

We intend this system intelligent following the definition given by McCarthy and Hayes [1].

'We shall say that an entity is intelligent if it has an adequate model of the world (including the intellectual world of mathematics, understanding of its own goals and other mental processes), if it is clever enough to answer a wide variety of questions on the basis of this model, if it can get additional information from the external world when it wants to, and can perform such tasks in the external world that its goals demand and its physical abilities permit.'

'A computer program capable of acting intelligently in the world must have a general representation of the world in terms of which its inputs are interpreted. Designing such a program requires commitments about what knowledge is and how it is obtained.'

Therefore it is possible to individuate two parts in this definition of intelligence, the epistemological and the heuristic one.

The first is the way of representing an external world in such a form that the solution of problems follows from the represented facts.

The second is the mechanism for solving problems on the basis of the informations expressed in the representation.

Many problem domains can be formalized, using the above mentioned definition, as a world with a set of actions which transform that world from one state to another. A particular problem is then specified by describing an initial state and a desired goal state. The problem solver is required to generate a plan, a simple sequence of actions which transforms the world from the initial state to the goal state.

Such problem domains include, but are not restricted to, applications to robot planning. [2] [3].

Intelligent robots, like the above described ones, have been realized in a lot of Research Centers in the last few years.

The methods used are different, but it is possible to individuate two important directions.

The first is a general purpose approach, characterized by the STRIPS programs family, based on formal logic [4]. They use a theorem prover in a goal oriented way. A more recent experience is the WARPLAN's one [5] and another evolution is the procedure generator LAWALY [6]. Among these systems only STRIPS is employed for a physical robot; the other ones are related to simulate robots.

The second method, introduced by Winograd [7] is to write procedures appropriate to the operators of the world. For a different world a new set of operators must be written, but the writing of the procedures is facilitated by the use of special languages, the goal oriented languages [8]. In this approach there are different recent systems for block manipulation and robot navigation [9] [10] [11].

In the problem solving field there are some different ways of representing knowledge in a notation which can be used in a computer program. Every representational scheme is based on the idea of separating individual entities and relationships between them.

From this point of view the predicate logic and the goal oriented languages are two substantially equivalent ways of representing the knowledge of the external world [12].

The choice among different ways of representation is justified by the richness of the schemes and their semantics, or by the programming facilities, or also by the exigence of a general purpose system [13].

For specified applications to the robot plan formation we propose the second approach, especially for its efficiency with respect to other existent systems.

The behaviour of a PLANNER - like language resembles the intuitionistic predicate calculus, but it is based on a particular way of the knowledge representation. This way, called procedural embedding [14] does not relate to the classical use of the logic for problem solving, and has been just inserted in the logical clauses interpretation [15].

In the procedural embedding the most significant aspect is that any piece of the knowledge can be represented by a procedure. The system uses the knowledge about the world by executing the procedures selected as relevant for a given task. Each procedure is identified by a pattern specifying the task for which the procedure can be used. All the procedures whose pattern matches the goal can be called; in this way non deterministic programs are obtained.

The mechanism for the procedure call, the particular structure of informations and the automatic backtracking are control informations, included in the language but related to its own deductive behaviour and not to world knowledge.

These innovations are introduced by the PLANNER of Hewitt [8] and they characterize the goal oriented languages.

### III - WORLD MODELING AND PLAN FORMATION

Our problem is simple enough but real. We want to control a robot which must move itself in an environment with obstacles and walls for accomplishing some tasks given by the operator, such as moving boxes in specified positions. Such a problem requires for the robot a non deterministic behaviour.

The problem proposed is significant because it is a real problem, it presents interesting characteristics and can be solved by the actual artificial intelligence techniques.

This problem has been yet studied and realized in some systems even if in more simple cases [4] [5] [6] [11]. We think such approaches are too much incomplete for real problems. It is possible to improve them, for example, by determining in a precise way the position of objects, the road for going to a position from another, and so on.

The project of the programming system for controlling this robot requires different steps at different detail's levels.

After having selected the procedural approach and defined in a precise way the physical space and the elementary actions of the robot, must be defined how those elementary actions have to be combined for achieving a task.

We choose to define these sequences of actions in a structured way, simple for the program comprehension and convenient for the robot.

We divide the program in procedures which make some changes in the world and in procedures which divide the problem in subproblems. These procedures are in a hierarchical order, so that one of them can call another of a lower level or itself.

The robot problems are of the sort that require quite complex and general world models compared to those needed in the solution of puzzles. Usually in puzzles and games, a matrix or list is adequate to represent a state of the problem. The world model for a robot

problem solver, however, needs to include a large number of facts and relations dealing with the position of the robot, and the positions and attributes of objects and boundaries.

Thus, the first question facing the designer of a robot problem solver is how to represent the world model.

Our system is analogous to the conventional ones, and it attacks the modeling of the world by dividing the knowledge in two categories:

i - one is the knowledge about the world at every moment, i.e. the knowledge of the objects and relationships among them.

ii - the other is the body of informations describing how the world may be transformed on the basis of the actions of the robot.

Those transformations are easily represented by a collection of operators. The operators are characterized by two principal components:

a - the first is a set of preconditions which must hold in the state of the world that the operator may be applied. The preconditions may contain variables which partially specify a state of the world. This partial specification defines a family of world states composing the domain of the operator.

b - the second is a set of effects eventually produced by the application of the operator, i.e. the set of the properties to delete and of the properties to add.

This theoretic representation can be illustrated by the simple following example.

We imagine a world consisting of a robot, two boxes B1 and B2, and a room. The robot and boxes may either be in the room or outside the room. A representation for one possible state of this world is the following:

(BOX B1)

(BOX B2)

(INSIDEROOM ROBOT)

(INSIDEROOM B1)

(OUTSIDEROOM B2)

The operations possible for this world are GOINTO-ROOM, GOOUTSIDEROOM, PUSHBOXIN, PUSHBOX-OUT, and some others.

The operator PUSHBOXIN may be defined:

name: PUSHBOXIN (robot pushes box b into the room)

parameters: b

preconditions: (OUTSIDEROOM ROBOT), (BOX b),  
(OUTSIDEROOM b)

effects:

delete: (OUTSIDEROOM ROBOT), (OUTSIDEROOM b)

add: (INSIDEROOM ROBOT), (INSIDEROOM b)

The operator may be applied if some b can be found which satisfies the preconditions; the value of b must by a box, this box must be outside the room.

For implementing such a description by the MICRO-PLANNER [16] we must translate it in procedural form, by using THASSERTIONS and THEOREMS.

(THASSERT (BOX B1))

(THASSERT (BOX B2))

(THASSERT (INSIDEROOM ROBOT))

(PUT 'PUSHBOX THEOREM ' (THCONSE (B) (INSIDEROOM (THV B))

(THGOAL (BOX (THV B)))

(THGOAL (OUTSIDEROOM (THV B)) (THTBF THTRUE))

(THGOAL (OUTSIDEROOM ROBOT) (THTBF THTRUE))

(THERASE (OUTSIDEROOM ROBOT))

(THERASE (OUTSIDEROOM (THV B)))

(THASSERT (INSIDEROOM ROBOT))

(THASSERT (INSIDEROOM (THV B)))

This theorem, of consequent type, is characterized by a pattern:

(INSIDEROOM (THV B))

which express the result of the application of the theorem, and by a sequence of instructions.

The theorem is written in a way such that 'the body implies the pattern'; if we want to demonstrate the truth of the pattern we must demonstrate, i.e. execute successfully, the body of the theorem. Every step of the theorem is an instruction, whose evaluation give 'success' or 'failure', as a result.

The possibility of accomplishing successive deductions is given because it is possible with those instructions to call other theorems.

When the problem to solve is expressed by the goal:

(THGOAL (INSIDEROOM B1) (THTBF THTRUE)))

the system researches by the pattern matching among the assertions and after among the theorems. If there are different assertions or theorems whose pattern matches the one of the goal the system makes an arbitrary choice, backing up and trying another automatically if one of them leads to a failure.

However this non determinism reduces the program efficiency and it is convenient to limit it as far as possible.

The solution proceeds normally in top-down or goal oriented way. It reduces the problems to subproblems, with the objective of reducing the original problem to a set of solved subproblems. There is also the possibility of bottom-up behaviour. In this case new assertions are derived from the old ones with the objective of deriving a solution of the original problem.

We can make the following observations comparing the theoretic representation with the program:

1. The parameters are the variables of the theorem; the theorem can have also other variables.
2. The pattern indicates that the operator PUSHBOXIN should be used only to achieve a goal INSIDEROOM b. The preconditions of the operators are satisfied by the goals: BOX (THV B), OUTSIDEROOM (THV B), and OUTSIDEROOM ROBOT.

Every goal can be obtained by searching in the assertions or by calling another procedure (by the form THTEF THTRUE is possible to call a procedure whose pattern matches the assigned one).

3. The add list and delete list are translated in the THassert and Therase form.
4. The order of the preconditions is a way for controlling the language and it is very important.
5. The set theoretic description of the operators does not include any indication of the strategy to be used in order to be able to apply an operator. In the MICRO-PLANNER language the program includes the following components: the order in which the elements of the precondition set must be satisfied, the task for which the theorem can be used (pattern), a backtracking monitor which will consider the different choices available to the system.

The result of the activation of the theorems is a plan for the robot. The plan produced consist of a list containing in their proper sequence all movements of the robot and blocks.

However, the world modelling schemes used by the robot system realized suffer from a number of limitations. The robot is a necessary participant in every operation, and it appears to have complete control over all changes in objects and relationships between objects. But in a larger world there may be many robots, each capable of causing changes in the world, and there exist a multitude of processes which transpire without the aid of robots.

Those conventional world modelling systems are also limited by their notion of time; the systems allow only one operation to be performed in any one instant. There is no notion of time duration associated with the operators.

The robot itself is easily confused; it cannot be given a second problem until it finishes solving the first, even though the two problems may be related in some way [17].

Another limitation is the linear solution in case of conjunction of goals; that is, the goals G1 & G2 can be solved in the given order or in the permuted order, but there is no facility of interleaving subplans.

These limitations are important for the representation theory but also in this approximation it is possible to solve many interesting problems.

#### IV - SYSTEM DESCRIPTION

Our system operates in a simulated environment. The world of our robot is a simple set of rooms described in a (x, y) coordinate system.

There are the robot, four pushable objects, walls and doors.

The primitive actions of the robot are:

- i. rolling forward a certain number of units in the direction of x and y axes
- ii. rotating 90° degrees in right and left direction
- iii. carrying or leaving some pushable objects.

Combining these actions the robot can travel around its world and it can move objects.

Even with such simple world, we can design and test quite complex hierarchical systems.

Our world model can be described by a set of assertions about the position of rooms and the location of various objects. This world is illustrated in figure 8.1.

The assertions for defining a room are of the type:

(THASSERT (A ROOM))

(THASSERT (A (0 6) (4 13))) whose meaning is 'A has values of the coordinates  $0 \leq x \leq 6$ ,  $4 \leq y \leq 13$ .

For defining a door:

(THASSERT (DOOR1 DOOR))

(THASSERT (DOOR1 (3 5) (4))).

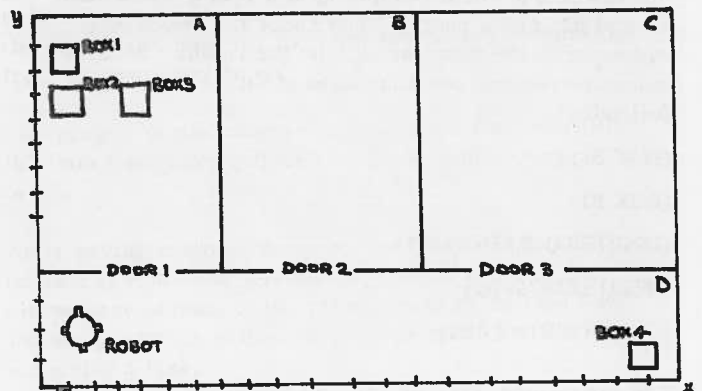


fig.8.1 - the world environment

For defining an object:

(THASSERT (BOX1 BOX))

(THASSERT (BOX1 PUSHABLE)).

These assertions are valid in all the program. The positions of the robot and the boxes at a specified moment are defined by some assertions of the type:

(THASSERT (ROBOT AT (1 2)))

(THASSERT (BOX1 AT (1 11)))

which are made or deleted by the actions of the robot.

We note that the greatest part of the assertions will not be changed by the application of operators.

The actions for the robot have to be organized in a convenient way for achieving some tasks. For this purpose we have implemented the procedures (called theorems in MICROPLANNER) here under described.

1. The PUSHBOX theorem is called by a goal which matches the pattern ((THV OBJ) AT (THV POS)).

The theorem is at the highest level and it is used for moving an object OBJ to a position POS. It can be applied with success only if OBJ is a pushable object, if POS is a physically defined position in which there are no object not pushable.

The problem is reduced into a set of subproblems, they are:

the robot must go to the actual position of OBJ,

the robot must carry OBJ,

the robot must go with OBJ to the final position POS,

the robot must leave OBJ in the final position.

Thus theorem does not modify the world model, but activates in a hierarchical way different theorems for solving the subgoals.

If there is a failure in one of these subgoals the theorem fails and the following actions are not executed.

2. GOROBOT has a pattern: (ROBOT AT (THV POS)). It is used for moving the robot with or without box to the position POS. The road for the robot is determined by another theorem at lower level, BREAKROAD. If such a road exists GOROBOT modifies the world model updating the assertions related to the robot position and eventually to the objects carried by the robot.

3. BREAKROAD is the first theorem for determining the road. Its pattern is: (PATH (THV OBJ) TO (THV POS)), and its task is to check the doors which the robot must cross, using DOOR theorem, and to break the road in subparts, one for each room in which the robot must cross.

It call one or some times the theorem PATH, which determines in a precise way the road between two points in the same room.

4. The pattern of the DOOR theorem is: (THV M) BETWEEN (THV I) (THV F)), and its task is to assign at M the value of the central point of the door between two rooms I and F.

5. The PATH theorem is called by a pattern (PATH (THV PI) (THV PF)) for constructing a path between PI and PF in the same room. If PI and PF are on the same axis, is called the theorem CONTROL which checks to see if there are any obstacles; else the theorem computes two new locations at the intersection of the x and y axes passing through PI and PF and calls the theorem CONTROL for the new paths so obtained.

If there are some obstacles the theorem uses two different strategies; ; if PI and PF are on the same axis, the theorem ROADRIGHT is called, which determines a new position out of the axis and calls again PATH between this new position and the final one.

- else a road until a suitable point before the obstacle is computed and PATH is called again between this new point and the final one.

6. The theorem CONTROL, whose pattern is (CONTROL (THV PI) (THV PF)), is called for checking if there are no obstacles between the positions PI and PF, which are on the same axis. For this task it calls either OBSTACLEX or OBSTACLEY.

7. The ROADRIGHT theorem is called if there is some obstacle between two points PI and PF on the same axis. The theorem determines a new point P, out of this axis, and calls again PATH between P and PF.

8. The theorems OBSTACLEX and OBSTACLEY, whose pattern is (OBSTACLE (THV PI) (THV PF)) find the obstacles between PI and PF. If such obstacles are invenied, the coordinates of the point in which the robot must stop and rotate are determined. The search is made so that the length of the path and the number of the rotations are minimized.

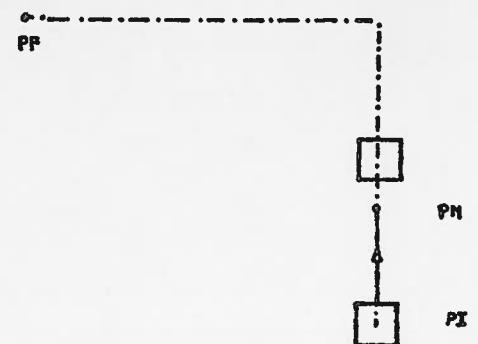
The examples illustrated in fig. 8.2 show this strategy:

- if the obstacle is on the first part of the path the point PN is determined as in fig. 8.2a.

- if the obstacle is on the second part the point PN is determined as in fig. 8.2b.

- if there are many obstacles some points PN are computed as in fig. 8.2c.

9. The theorems CARRY and LEAVE are called for modifying the world representation when the robot carries or leaves an object.

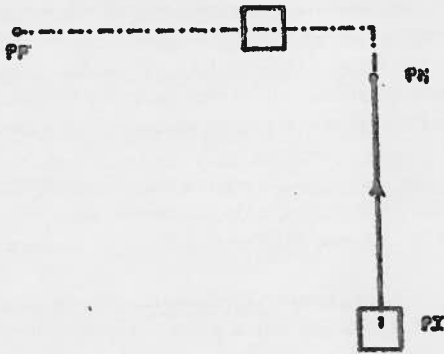


8.2a.

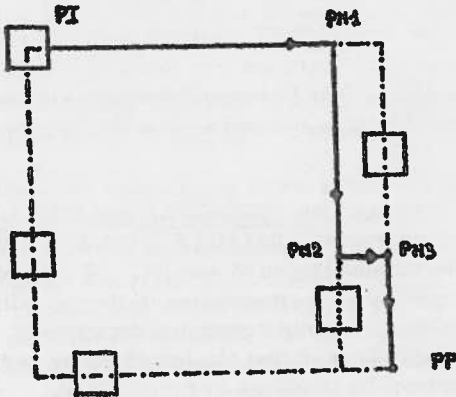
10. The theorem OCCUPY is an antecedent theorem, i.e. a theorem called when an assertion is made which matches its pattern ((THV OBJ) AT (THV POS)). This theorem inserts in the data base a new assertion ((THV POS) OCCUPIED)).

In Appendix there is the sequence of steps for achieving the following goals:

(BOX1 AT (3 3))



f.2b.



f.2c.

fig. 8.2 - strategies for determining the road when there are obstacles.

(BOX4 AT (3 9))

(BOX3 AT (7 8))

We illustrate in fig. 8.3 the solution of the first goal. The subgoals have a number which indicates the order in which they are activated. The various levels are

expanded in a depth-first manner. We note that the goals illustrated are only the activated ones with success; between the 18th and the 19th goal there is a path with failure, represented in fig. 8.4.

In fig. 8.5 there are three different states of the world in relation with the solution illustrated in fig. 8.3.

V - CONCLUDING REMARKS

The preceding sections have outlined the methodology for the construction of the world model and for the plan formation with the goal oriented languages.

Some characteristics of the proposed program are now focused.

The knowledge represented by the assertions is very reduced and the new knowledge useful is memorized as soon as it is computed. For example, the doors are defined only by their coordinates; when the robot must go in a room, it selects the door between the two rooms, and the system memorizes this knowledge in a suitable assertion. If the task given to the robot does not require to cross some doors, this knowledge is not necessary. This way of keeping some partial results, again useful, is very advisable in large systems.

The program is not deterministic, but in a limited way. Many parts are programmed in LISP, because they do not require the backtracking. Thus the efficiency of the system is improved.

There are many possibilities in the determination of the paths; but the top-down activation reduces the width of the goal-tree.

The manipulations of the assertions are few, and they are not influenced by the backtracking since they are made only in one theorem of high-level, after the success of the low-level goals.

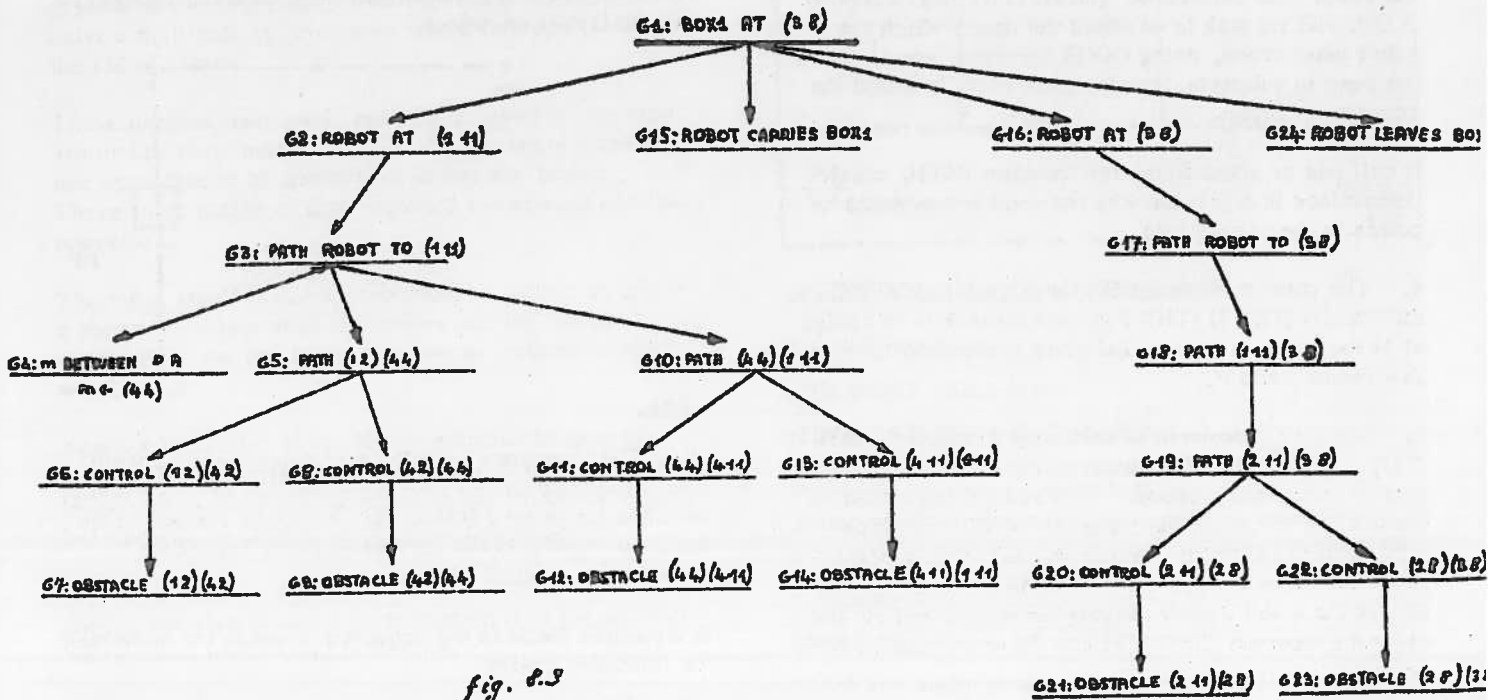


fig. 8.3

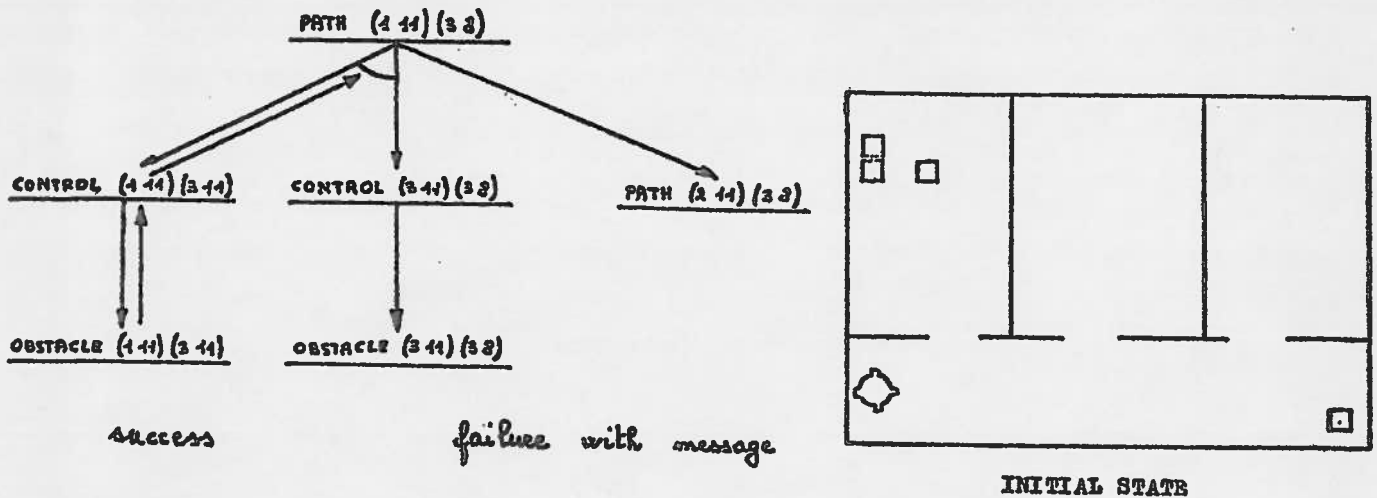
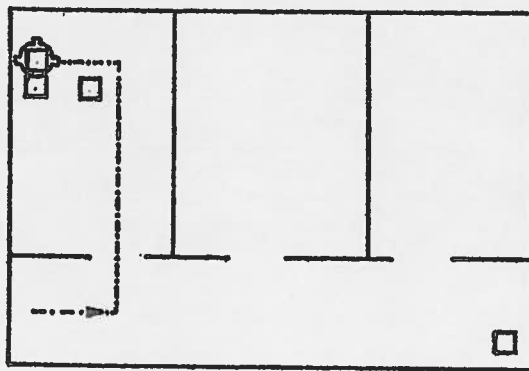
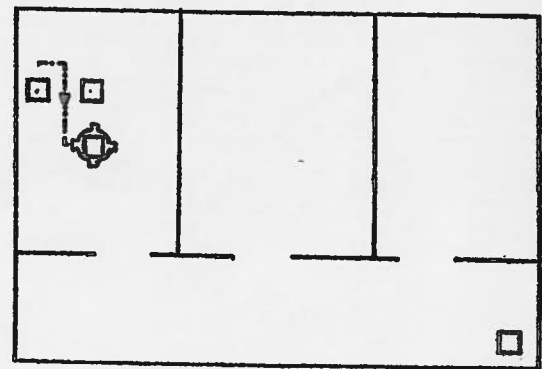


fig.8.4 - failure after the 18th goal.



STATE AT THE 15th GOAL



FINAL STATE

fig.8.5

The system is easy extendible; it is possible to add new features to the general structure of the program for solving new tasks merely by writing new theorems.

Extensions of this system can introduce in it a more complex world model.

We think to introduce the time notion, for organizing a set of goals in a more real way. For example, if there is a task which does not require the continuous presence of the robot, as filling a bucket with water, the robot can begin a new action before the accomplishment of the precedent one. It should be also convenient to give to the robot the possibility of organizing a given set of goals in a convenient order, or to coordinate several robots interacting in the same world.

## REFERENCES

1. McCarthy, J., Hayes, P.J. 'Some Philosophical Problems from Standpoint of Artificial Intelligence', Machine Intelligence, Vol.4, American Elsevier Publishing Company, New York, 1969.
2. Nilsson, N.J. 'Problem Solving Methods in Artificial Intelligence', McGraw-Hill Book Company, New York, 1971.
3. Fikes, R.E., Hart, P.E., and Nilsson, N.J. 'Some New Directions in Robot Problem Solving', B.Meltzer & D.Michie, Eds., Machine Intelligence, No.7, Edinburgh University Press, Edinburgh, 1972.
4. Fikes, R.E., Nilsson, N.J. 'STRIPS: a New Approach to the Application of Theorem Proving to Problem Solving', Proc.2nd IJCAI, London, 1971.
5. Warren, D.H.D. 'WARPLAN: a System for Generating Plans', DCL Memo.76, University of Edinburgh, Edinburgh, 1974.
6. Siklossy, L., Dreussi, J. 'An Efficient Robot Planner which generates its own Procedures', Proc. 3rd IJCAI, Stanford University, Stanford, California, 1973.
7. Winograd, T. 'Procedures as a Representation for data in a Computer Program for Understanding Natural Language', A.I.Lab., AI TR-17, MIT, Cambridge, Massachusetts, 1971.
8. Hewlett, C. 'Description and Theoretical Analysis (using schemata) of PLANNER: a Language for Proving Theorems and Manipulating Models in a Robot', Ph.D. Thesis, MIT, Cambridge, Massachusetts, 1972.



9. Fahlman, S.E. 'A Planning System for Robot Construction Tasks', A.I. Lab., AI TR-283, MIT, Cambridge, Massachusetts, 1973.
10. Derksen, J., Rulfson, J.F. and Waldinger, R.J. 'The QA4 Language Applied to Robot Planning', A.I. Center, Tech. Note 65, SRI, Menlo Park, California, 1972.
11. Nilsson, N.J. 'A Hierarchical Robot Planning and Execution System', A.I. Center, Tech. Note 76, SRI, Menlo Park, California, 1973.
12. Hayes, P.J. 'Some Problems and Non-Problems in Representation Theory', Proc. AISB Summer Conference, University of Sussex, England, 1974.
13. Corti, P.L., Gini, G., Gini, M., and Pagello, E. 'L' Impiego della logica matematica e dei linguaggi orientati ad obiettivi nei problemi di robotica', MP-AIM, Istituto di Elettrotecnica ed Elettronica del Politecnico di Milano, Milano, 1974.
14. Hewitt, C. 'Procedural Embedding of Knowledge in PLANNER', Proc. 2nd IJCAI, London, 1971.
15. Kowalski, R. 'Logic for Problem Solving', DCL Memo 75, University of Edinburgh, Edinburgh, 1974.
16. Sussman, G.J., Winograd, T., and Charniak, E. 'MICROPLANNER Reference Manual', A.I. Lab., AI TR 203, MIT, Cambridge, Massachusetts, 1970.
17. Hendrex, G.G. 'Modeling Simultaneous Actions and Continuous Processes', Artificial Intelligence, 4, 1973.

## How many Industrial Robots are there?

### Ten, Fifty, Hundred....



Complete data on 138 industrial robots manufactured in the U.K., Japan, Europe and U.S.A. are contained in a 176 page report "Industrial Robots - A Survey". Application trends generally and in automatic warehouses in particular are discussed. Basic principles explained and background information provided.

Price: £12.50 U.K., \$ 35.00 elsewhere (Airmail postage \$ 5.00 extra)

To: International Fluidics Services Ltd. Carlton, Bedford, England

Please supply..... copy/copies of Industrial Robots - A Survey

NAME: \_\_\_\_\_

ORGANISATION: \_\_\_\_\_

ADDRESS: \_\_\_\_\_