Automation and Robotics in Construction XI
D.A. Chamberlain (Editor)
1994 Elsevier Science B.V.

AN INTELLIGENT TASK-PLANNING SYSTEM FOR AUTONOMOUS INTERIOR-ROBOTS

Igal M. Shohet, Yehiel Rosenfeld and Abraham Warszawski

National Building Research Institute
Technion - Israel Institute of Technology
Haifa 32000, ISRAEL

ABSTRACT

Autonomous task-planning is a major phase within the systemic approach towards robotized performance of interior-finishing building tasks. This paper presents mathematical formulations and offers near-optimal solutions for three hierarchical levels of the task planning procedure: The macro-level deals with the robot's travelling among multiple rooms on a building floor. This is determined by the application of the well known "Travelling Salesman's Problem (TSP)" algorithm, in which each node on the network represents a room, and each arc represents a door between two rooms. The next main level deals with the near-optimal positioning and routing of the robot among workstations within a room. An original algorithm was developed to minimize the total cost through dynamic programming by a recursive solution. The last, micro-level of pre-planning the exact path of the tool from each workstation is briefly presented.

1. INTRODUCTION

TAMIR - Technion Autonomous Multipurpose Interior Robot - has been developed at the Israel Institute of Technology in a multiphase, systemic R&D project. It has been described comprehensively in Rosenfeld, Warszawski and Zajicek (1993), Warszawski and Rosenfeld (1994), and in several ISARC symposia proceedings. The project has been aimed at examining - both theoretically and practically - the feasibility of performing interior building tasks with the aid of robots. This pre-prototype building robot consists of an articulated arm with six degrees of freedom and a reach of 1.7 m, mounted on a computer-controlled three-wheeled carriage measuring 0.85x0.85 m. The general concept of operation assumes that prior to performing the task - e.g. painting, plastering, tile-setting or the building of partitions throughout a building floor - the robot would already have a complete map of the floor layout within its operating system, and a preplanned path to advance from room to room and inside each room. The carriage - equipped with automatic navigation devices - serves as a moving platform for the robotic arm, taking it from one *workstation* to the next, allowing the arm to perform its task in the vicinity of the particular workstation, usually for a duration of 3-20 minutes at each. The carriage is also equipped with four stabilizing jacks, which can be deployed and levelled automatically at the workstation prior to commencing the task performance.

This paper focuses on one aspect of this systemic approach: Autonomous task-planning for an interior finishing robot. A method has been developed for planning of an efficient sequence of the robot movements from room to room, as well as the sequence and the exact positioning of workstations within each room. The task-planning procedure uses map files of the array of rooms that form the floor, created either manually or by the robot in the previous phase of floor-mapping. The task-planning procedure finds the best route (i.e. the shortest in time) to perform all the work required on the entire floor and creates new navigation files that will direct the robot in the next phase of autonomous task-performance. The computing time for one room is short enough (less than one minute) to permit *real-time* adjustments of the plan to minor changes that may be discovered by the robot upon entering a room through a quick scanning of the room's contour.

2. GENERAL DESCRIPTION OF THE TASK-PLANNING PROCEDURE

The task-planning procedure, like the floor mapping procedure that precedes it (described in Warszawski, Rosenfeld and Shohet 1992) utilizes several geometric characteristics with regard to the shape of common rooms in buildings. The following assumptions greatly simplify the mathematical procedures, and shorten the execution time of the algorithms:

(1) The floor layout is orthogonal, i.e. each wall is perpendicular to its adjacent walls.
(2) All walls are perpendicular to the floor and to the ceiling.
(3) All surfaces (floors, walls and ceilings) are planar.

If some spaces/rooms do not conform to these assumptions, they will simply be marked as unidentifiable spaces and skipped in the robotic task-planning, to be performed either manually or semi-automatically with close human intervention.

The task-planning procedure consists of three hierarchical levels:

(1) The first, the macro level, uses a modified version of the well known "Travelling Salesman's Problem" to determine the optimal sequence of "visiting" the various rooms (including corridors) presented as a two-dimensional array of interconnected orthogonal spaces (not necessarily rectangles), some of them with more than one entrance.
(2) The second, the main level, uses "dynamic programming", to determine, through a recursive process, the optimal positioning of the workstations within each room so as to minimize the total cost of completing the entire task. (For practical considerations it was decided that once the robot enters a room it would not leave it without completing the task in it; in other words, it will not enter through one entrance, paint a part of the room and exit through another entrance without completing the whole task in the room.)
(3) The third, the micro level, determines the precise execution of the task at each workstation, essentially the exact path of the tool, such as a spraygun, over the limited work-scene covered from that workstation.

The entire procedure, written in "Turbo Pascal-6", has been tested with the task of wall spraying, through graphic simulation imitating the robot's movements and the spraying of walls throughout an entire building floor, while omitting predefined areas - such as doors and windows.

The computing time of near-optimal workstation locations within a single room only rarely exceeds one minute when an IBM 486, 33 MHz personal computer is used. This performance can be further enhanced by better hardware. Such a rapid procedure permits a very efficient combination of robotic floor-mapping with autonomous task planning: While the robot moves from room to room during the mapping procedure, the data of the last room can be processed by the computer for the task-planning procedure. Thus, upon the return of the robot from its "voyage" of autonomously mapping an entire floor, the printer can "hand" to the operator a detailed plan for the autonomous execution of the task by the robot. In addition to the planned path of the robot and the precise locations of the various workstations, this plan may also include a near-accurate time estimate for the execution of the job, as well as a complete list of quantities of materials required for the job, and the desired places at which to pile them along the route....

3. DEFINITION OF THE TASK-PLANNING PROBLEM

Interior finishing activities in buildings can be categorized from various aspects. One relevant division for task-planning purposes distinguishes between *horizontal surfaces* and *vertical surfaces*. The geometric characteristics, and the consequent task-planning procedures of these two classes, differ from each other. For coating tasks - such as painting, plastering and tiling - the "horizontal" category requires full coverage, of a single room or an array of rooms all on a common plane with a specific application. From each workstation, the robot is capable of covering a typical geometric shape, determined by the robot's effective work envelope for the particular task with the particular tool (e.g. for painting ceilings, the robot is capable of spraying from each workstation, a typical circular area on the surface of the ceiling). The "vertical" category - dealt with in this paper - is far more complicated to formulate: The "footprint" of the walls on the floor is presented by orthogonal, yet irregular, closed polygon shaped spaces, called "rooms", which are interconnected through openings (discontinuities of the lines) called "doors". Rooms are sub-entities of floors; walls are sub-entities of rooms; and doors and windows are sub-entities of walls.

The robot has to move along the contours of each room in relatively small steps and, from each workstation, to cover a certain linear length of the wall/walls, until the entire task is done. The length of the wall section/s that can be covered from each workstation is determined by the spatial intersection of the wall surface represented by the side of a polygon with the effective work envelope of the robot, for a particular task with a particular tool. The next workstation will usually be located further along the wall, in such a manner that the robot would be capable of continuously covering maximum additional surfaces of the wall/s.

In a confined room, and especially in smaller rooms whose shape is not a simple rectangle, there are multiple possible combinations of workstations which, while all provide full coverage of the walls, may differ substantially both in the total number of workstations required and in the total time spent by the robot for straight-line travelling, for turns, for set-up and levelling at each workstation, etc.

Additionally, there are many alternative sequences for moving from room to room in order to visit all of them. Again, the alternatives will differ in their total travelling time.

The purpose of the optimization algorithms is to try to minimize these times without completely calculating each and every one of the multitude of possible alternatives.

4. REPRESENTATION OF THE ROBOT'S WORK ENVELOPE

The *nominal work envelope* of a robot is defined as the collection of all points in space (around the robot) that its arm end ("bare-hand") is capable of reaching, regardless of its orientation.

The *effective work envelope* for a particular task with a particular tool is defined here as the modified nominal work envelope, which also takes into consideration the addition of the particular tool and the required orientation of the tool (and sometimes also its distance) with respect to the treated surface. It was first treated in the early phases of this project by Argaman (1989). In spraying, for example, the spraygun must "shoot" in a nearly perpendicular orientation with respect to the treated wall, from a prespecified distance.

Fig. 1 shows the nominal work envelope of the GMF-S700 robot, while Fig. 2 presents its modified effective work envelope for wall painting with a particular spraygun. The dotted line on Fig. 2 shows the modification of the external nominal boundaries, for a spraygun mounted on the end of the arm shooting horizontally towards the vertical surface of the wall, while at a distance of 0.30 m from it. It also shows the same modifications for the internal surface of the work envelope. The height of the wall, and the allowable angle of deviation from horizontal shooting (in this case 20°), finally determine the effective external and internal radii (R_{ext} and R_{int}) around the robot's center that create the shape of a washer (shaded on Fig. 2). From each workstation, the robot is capable of treating the wall all the way from the floor to the ceiling, only for those sections that are included in the space of the "hollow cylinder", presented on the floor plan by this shaded "washer".

In summary, given the geometric features and dimensions of the robot, the tool, the wall, and the relations between the tool and the wall, it is possible to "reduce" the complicated spatial intersections to the simple projection form of a "washer" on the floor movable along the contours of the room and "stamping" them until they become fully covered with "stamps". Each "stamp" represents a workstation, and the objective is to minimize not merely the number of "stamps", but also the total travelling time among them, as formulated mathematically in the next sections.

5. PRINCIPLES OF THE MATHEMATICAL SOLUTION

The flowchart in Fig. 3 presents the procedure of the solution:

Stage 1 is performed through the application of a TSP (e.g. Golden and Assad, 1986) algorithm, in which each floor is represented as a network composed of rooms represented as nodes, and doors – as arcs. Due to space limitations, this algorithm cannot be presented here.
Stage 2 is a feasibility check in each room for determining whether all walls are accessible to the robot, and deleting from the task-planning those sections that are not accessible.
Stage 3 determines a near-optimal arrangement of workstations within each room using dynamic programming in a recursive process.
Stage 4 evaluates the order of travelling among the workstations determined in stage 3, to provide complete coverage in minimum time.
Stage 5 is a micro-planning of the path of the tool from each workstation, based on Zajicek (1992).

The most substantial part in the optimization process is stage 3 – the positioning of workstations in each room.

6. DETAILS OF THE MATHEMATICAL SOLUTION

The representation of the effective work envelope by a simple "washer", can be reduced even further, to take the shape of the letter "T" as presented in Fig. 4. The proportions among the "leg" and the "wings" of various "T"-like alternatives (noted as D and L respectively), that can be fit into the "washer", can be calculated by simple trigonometric formulae. D_i should be between R_{int} and R_{ext}, and L_i will be determined accordingly (or vice-versa). In a corner situation, for example, the robot may cover a section with the length L_1 of one wall of that corner and L'_1 of the other, or, instead, L_2 of the former and L'_2 of the latter (see Fig. 4).

The optimization procedure finds the best combinations, considering a certain number of successive steps (for practical purposes 4 to 6) to cover the entire wall-layout of the room.

The following solution is based on the Bellman optimality principle (Bellman and Dreyfus, 1962) using dynamic programming (e.g. Dreyfus and Law, 1977).

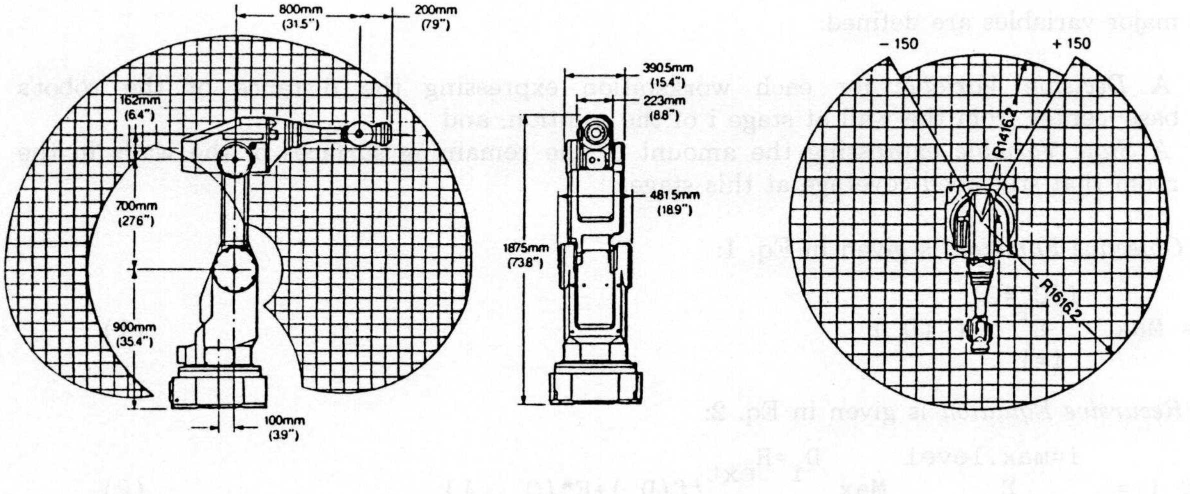


Fig. 1: The nominal work envelope of the GMF-S700 robot.

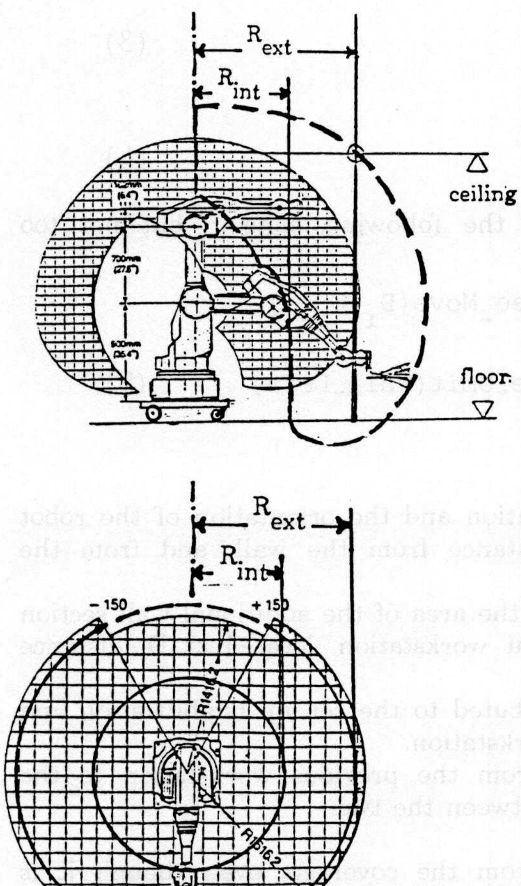


Fig. 2: The modified effective work envelope for wall painting.

1. Determine the order of rooms to be performed based on minimum vehicle travelling among them.
2. Perform a feasibility check for each room - mark the wall sections accessible to the robot.
3. Allocate exactly the work-stations in each room, using the criterion of minimal stations.
4. Determine the sequence of workstations in each room according to entering door and least-cost of vehicle movements.
5. Divide each workstation to sub-sections and determine the exact path of the tool in each of them.

Fig. 3: Flow-chart of the hierarchical autonomous task-planning.

Two major variables are defined:

D_i – A *Decision Variable* for each workstation expressing the distance of the robot's base-center from the wall at stage i of the solution, and
C_i – A *State Variable* expressing the amount of the remaining surfaces of the walls in the room that still need coverage at this stage.

The *Objective Function* is given in Eq. 1:

$$F^* = \text{Max} \sum_{i=1}^{i(C_i=0)} f(D_i) \tag{1}$$

The *Recursive Equation* is given in Eq. 2:

$$F^*(C_i) = \sum_{i=1}^{i=\text{max.level}} \underset{D_i=R_{int}}{\overset{D_i=R_{ext}}{\text{Max}}} \{f(D_i)+F^*(C_{i+1})\} \tag{2}$$

The *Step Increments*, D_{step}, between the discrete values of D_i through the recursive process, should be defined arbitrarily, and it is suggested to divide the distance between R_{ext} and R_{int} into 3 to 5 sections, namely:

$$D_{step} = D_{i+1}-D_i \approx \frac{R_{ext} - R_{int}}{4} \tag{3}$$

The *transition equation* from stage to stage is given by:

$$C_{i+1} = C_i - \text{Cover}\ [\text{Loc}(D_i,C_i)] \tag{4}$$

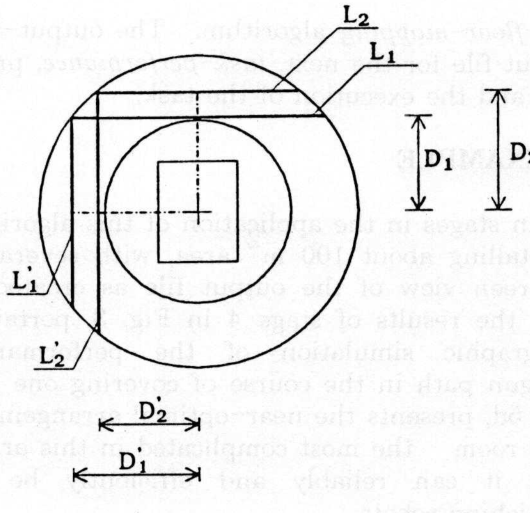
And, finally, the *Utility Function* is given by the following, long – but not too complicated – expression:

$$f(D_i)=\text{Cover}[\text{Loc}(D_i,C_i)]-\text{Fee\_Station}(D_i,C_i)-\text{Fee\_Move}(D_i,C_{i-1}) -$$
$$\text{Fee\_Turn}[\text{Loc}(D_i,C_i),\ \text{Loc}(D_{i-1},C_{i-1})]-\text{Fee\_Omit}(\text{Wall\_Left}) \tag{5}$$

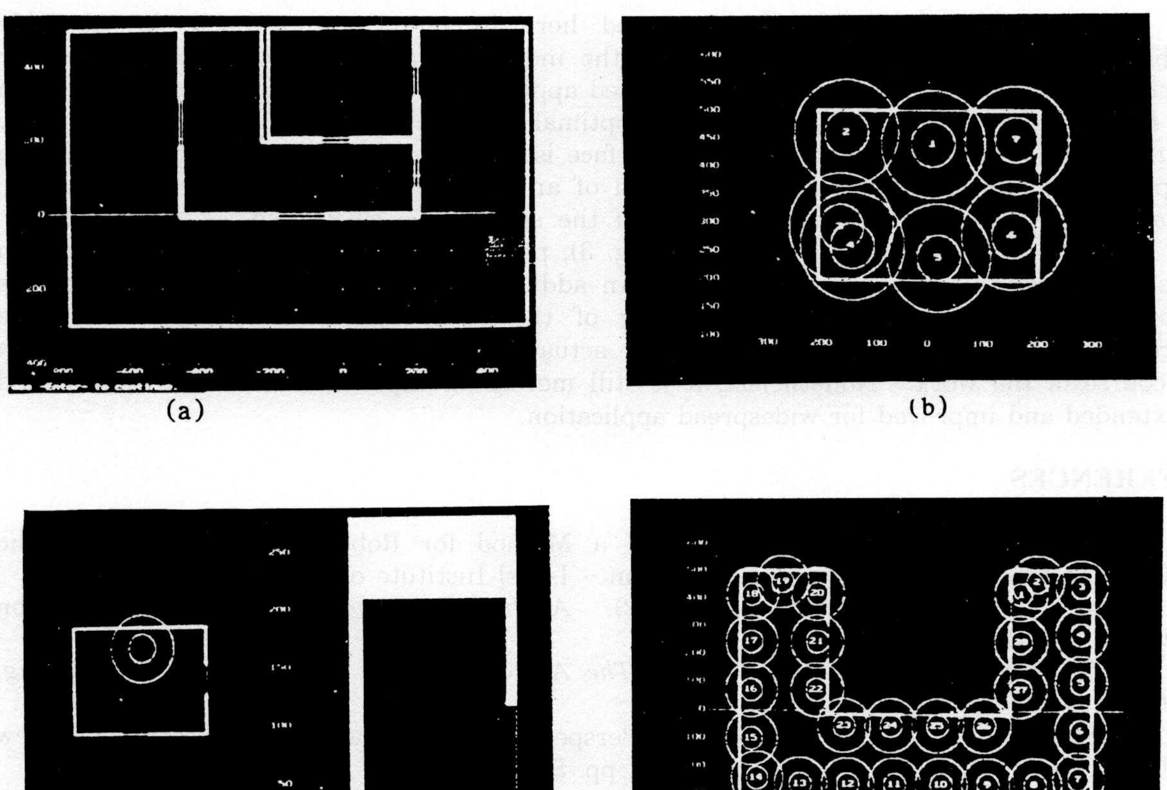
Where:

$\text{Loc}(D_i,C_i)$	– a function that returns both the location and the orientation of the robot workstation derived from its D_i distance from the wall, and from the remaining surface to be covered – C_i.
Cover	– a function that returns the length or the area of the additional wall section that can be covered from the present workstation, located at D_i distance from the wall, in stage C_i.
Fee_Station	– a fee (in terms of time or cost) attributed to the set-up (stabilization and calibration) required at every new workstation.
Fee_Move	– a fee attributed to the travelling from the previous workstation to the present, dependent on the distance between the two.
Fee_Turn	– a fee for each turn along the route.
Fee_Omit	– a fee for omitting a wall section from the coverage, even though it is accessible by the robot according to the initial feasibility check.

The last case may mainly happen in the vicinity of corners, where (especially in the absence of this fee), the algorithm may prefer leaving some small areas untreated instead of assigning a special workstation for just a small piece of wall. This entire algorithm has been tested on various cases and different wall layout arrangements, according to the stages of Fig. 3. The initial input for the *task-planning* algorithm is, as mentioned above, the output



<u>Figure 4</u>: Intersection of the simplified work envelope with the wall in a corner.



<u>Figure 5</u>: Graphic simulation for task-performance to the autonomous task-planning for an array of three rooms.

a. Layout of walls.
b. Arrangement and sequence of workstations in the first room.
c. Micro-planning of the path of spraygun on the wall.
d. Arrangement and sequence of workstations in the third room.

file created in its preceeding *floor-mapping* algorithm. The output-file of this algorithm, in turn, should serve as the input file for the next, *task-performance*, program that controls the actual travelling of the robot and the execution of the task.

7. A DEMONSTRATIVE EXAMPLE

Fig. 5 presents the main stages in the application of this algorithm on a small array of non-trivial shaped rooms totalling about 100 m^2 area, with several inter-connecting door openings. Fig. 5a is the screen view of the output file as created by the floor-mapping algorithm. Fig. 5b presents the results of stage 4 in Fig. 3, pertaining to the first room; while Fig. 5c presents graphic simulation of the performance according to the micro-planning of the spraygun path in the course of covering one of the walls in the first room. The last picture, Fig. 5d, presents the near-optimal arrangement and the sequence of the workstations in the third room - the most complicated in this array. Many test-runs of the algorithm proved that it can reliably and efficiently be used for autonomous task-planning for interior finishing robots.

8. CONCLUSION

The task-planning approach, presented here in essence, appears to be a viable candidate, to be adopted as the standard of the industry for autonomous task planning for interior building robots (as well as other related applications). It provides a feasible solution for full coverage; it guarantees a near-optimal solution; it is fully automated; the calculations are very rapid; and the user interface is fairly simple. The algorithm provides a comprehensive solution, from the macro-plan of an entire floor (e.g. of a large residential, office or hotel building) to the micro-plan of the spraygun path around a specific window. The hierarchical nature of the algorithm (Fig. 3), permits efficient computer time allocation either by serial or by parallel processing. An additional advantage is the flexibility of the system that allows fast and easy updating of the original plan in real-time, if some discrepancies between the theoretical and the actual geometry of the room are discovered in the course of the work. Nonetheless, it is still merely an experimental version that should be extended and improved for widespread application.

REFERENCES

1. Argaman, H. (1989). "Development of a Method for Robotization Planning on the Building Site", D.Sc. Dissertation, Technion - Israel Institute of Technology.
2. Bellman, R.E. and Dreyfus, S.E. (1962). *Applied Dynamic Programming*, Princeton University Press, Princeton, N.J.
3. Dreyfus, S.E. and Law, A.M. (1977). *The Art and Theory of Dynamic Programming*, Academic Press, New York, N.Y.
4. Golden, B. and Assad, A. (1986). "Perspective on Vehicle Routing: Exciting New Development", Operations Research 34(5), pp. 803-810.
5. Rosenfeld, Y., Warszawski, A. and Zajicek, U. (1993). "Full Scale Building with an Interior Finishing robot", *Automation in Construction*, Vol. 2(4), pp. 229-240.
6. Warszawski, A. and Rosenfeld, Y. (1994). "Robot for Interior Finishing Works in Building - Feasibility Analysis", *Journal of Construction Engineering and Management, ASCE (American Society of Civil Engineers*, Vol. 120(1).
7. Warszawski, A., Rosenfeld, Y. and Shohet, I.M. (1992). "An Autonomous Control System for the Interior Finishing Robot", Proc. of the 9th ISARC Conf. for Robotics & Automation in Construction", Tokyo, Japan.
8. Zajicek, U. (1992). "Development of Building Applications in Robotized Technology", M.Sc. Thesis, Technion - Israel Institute of Technology.