

1 of 1

SANDIA
CONF-9311103-1

2

Motion Planning for Robotic Spray Cleaning with Environmentally Safe Solvents *

Yong K. Hwang, Lilita Meirans and William D. Drotning
Sandia National Laboratories
Albuquerque, New Mexico 87185, USA

ABSTRACT

Automatic motion planning of a spray cleaning robot with collision avoidance is presented in this paper. In manufacturing environments, electronic and mechanical components are traditionally cleaned by spraying or dipping them using chlorofluorocarbon (CFC) solvents. As new scientific data show that such solvents are major causes for stratospheric ozone depletion, an alternate cleaning method is needed. Part cleaning with aqueous solvents is environmentally safe, but can require precision spraying at high pressures for extended time periods. Operator fatigue during manual spraying can decrease the quality of the cleaning process. By spraying with a robotic manipulator, the necessary spray accuracy and consistency to manufacture high-reliability components can be obtained. Our motion planner was developed to automatically generate motions for spraying robots based on the part geometry and cleaning process parameters. For spraying paint and other coatings a geometric description of the parts and robot may be sufficient for motion planning, since coatings are usually done over the visible surfaces. For spray cleaning, the requirement to reach hidden surfaces necessitates the addition of a rule-based method to the geometric motion planning.

1. INTRODUCTION

As the use of CFC solvents is discouraged and tightly controlled due to their destructive effects on stratospheric ozone, more industries are using aqueous solvents to clean electronic and mechanical components. Aqueous solvents, however, may be less effective than CFC and need to be sprayed over the surfaces at a high pressure for a longer period of time. This mandates the use of a robot since the spray gun is heavy, operates at high pressure, and precision and consistency are required for cleaning.

Planning robot motions for spray cleaning is a time-consuming task. Different motions must be programmed for different types of components. Even for the same component, the motion parameters

have to be modified according to the desired cleanliness, degree of contamination of the component, and the cleaning process parameter variations such as the rate of chemical reaction. Since manually programmed motions cannot be easily modified, there are significant delays and costs whenever the components or cleaning process parameters change.

Currently, motions of most robots are programmed using a teach pendant or off-line simulation software such as CimStation, IGRIP or ROBCAD. None of these methods offers a convenient way to modify planned motions. In using simulation software, the user defines a sequence of coordinate axes which is then tracked by the coordinate axes of the spray gun. Although more convenient than using a teach pendant, placing coordinate axes in the 3-dimensional space from the 2-dimensional screen requires a lot of training and time. There are painting motion planners in CimStation and IGRIP [9, 10] to alleviate this problem, but these planners can only handle more or less flat surfaces. There is also a painting planner that generates spray sweep patterns [8], but it is also intended for flat or slightly curved surfaces. The only existing motion planners with collision avoidance capability are the ones that move robots from one point to another, i.e., robot tool tips are not required to trace any curves [1, 2].

This paper presents two motion planners that automatically generate motions for spray cleaning robots based on the component geometry and a cleaning process model; they are a geometry-based planner and a rule-based planner. The geometry-based planner computes paths for the spraying robot using only geometric information about fully 3-dimensional components. The computed path has the property that all points on the surfaces of the components are covered by the spray cone for a required time duration, the amount of solvent used is minimal, and the robot manipulator does not collide with components or fixtures during the motion. This planner does not model the solvent flow over hidden surfaces, chemical reactions, or the movement of dirt particles over the surfaces. Instead, a rule-based planner has been developed to address these problems. Rules were developed for the cleaning process based on solvent flow and dirt movement from empirical observation, and spray motion is planned using these rules. The geometry-based planner is useful for individual components whose surfaces are not hidden, whereas the rule-based planner is appro-

*This work has been performed at Sandia National Laboratories and supported by the U.S. Department of Energy under Contract DE-AC04-76DP00789.

11/11/93

appropriate for assembled components such as electronic circuit boards with hidden surfaces.

In the actual spray-cleaning robot system, spray motions are designed in two stages. First, the CAD model of a component and the cleaning process parameters are used to plan a robot motion, and the cleaning process is simulated. If the surface cleanliness in the simulation is adequate, the planned motion is executed on the real robot and the component. If the component is not cleaned thoroughly, the cleaning process parameters are adjusted and the simulation is used again to plan robot motion. This process is repeated until a motion thoroughly cleaning the component is obtained. This motion is then stored and used in a production environment.

This paper is organized as follows. Section 2 describes the geometry-based planner, and the rule-based planner is explained in Section 3. The performance of each planner is illustrated in Section 4, and the conclusions are in Section 5.

2. GEOMETRY-BASED PLANNER

The geometry-based planner plans motions that will cover all points on the surface with the spray cone while avoiding collisions between the robot and the components. This planner assumes that only points visible from the spray nozzle tip are cleaned, and thus does not consider cleaning effects of hidden surfaces due to solvent flow. Components are modeled as polyhedra, and are represented using a commercial solid modeler ACIS. Although ACIS can handle curved objects, our planner is currently limited to polyhedral objects.

The planner generates spray motion in four steps. First, the surfaces to be cleaned are partitioned into stripes of width equal to the effective width of the spray cone. The centerlines of the stripes will be traced by the center of the spray cone (see Figure 1). Second, the stripes are ordered to minimize the total traveling distance by the spray gun, which includes both traveling along the stripes and traveling from one stripe to the next. Third, a curve-following routine is used to make the center of the spray cone trace the centerlines of the stripes, while avoiding collisions between the robot links and components and fixtures in the work space. Lastly, a cleaning analysis routine computes the cleanliness of each point on the surfaces to verify that the component is adequately cleaned. Each of these steps is explained below.

2.1 Striping Routine

There are many ways to partition a face of a polyhedron into a set of stripes. Since the width of stripes is set to the width of the spray cone only the stripe direction needs to be determined. Important criteria for the stripe direction are the preferred motion direction of the robot and the number of resulting stripes. In the current system, we set the stripe direction along the longest edge of the face, since this heuristic tends to minimize the number of the stripes. Other striping methods can be easily incorporated in the system later on.

2.2 Ordering of Stripes

Once all the surfaces are striped, the stripes need to be ordered so that the total cleaning time is minimized. To get a true optimal path, we must consider the robot kinematics and dynamics, which complicates an already complex problem. We instead minimized the total distance traveled by the center of the spray cone and assume that this measure will give a near minimum-time path of the robot. A path tracing a set of stripes consists of two kinds of subpaths: the paths along the stripes and the paths used to move from one end point of a stripe to that of another. This stripe ordering problem is exactly the *rural postman* problem, which is a variation of the traveling salesman problem [6].

Our stripe ordering problem, however, has a special structure in which stripes are grouped according to the faces they belong. Since the angle of the spray gun usually changes significantly when moving between faces, and stripes from the same face are situated closely together, it is a good heuristic to completely clean one face before cleaning another. We therefore solve the stripe ordering problem in two stages. First, we solve the rural postman problem once for the set of stripes associated with each face. Then we make an edge connecting the entry and exit points of the spray gun for each face, and set the edge cost equal to the length of the solution path. With these edges, we then solve another rural postman problem to get an optimal ordering of the faces. This two step approach has proven to be near optimal for most components, and conformed closely to human intuition.

2.3 Curve-Following Routine

There are many algorithms that make a robot tool tip trace a curve while optimizing some performance measure such as obstacle avoidance and singularity avoidance [4, 5, 7]. We have modified the algorithm in [7] so that the center of the spray cone follows the centerlines of stripes as closely as possible, and the spray angle is as perpendicular to the surface as possible. The spray angle is negotiated only if necessary for collision avoidance. In the inverse kinematics computation, we have used the pseudo-inverse of the manipulator Jacobian.

2.4 Cleaning Analysis

After the spray cleaning motion is planned, we simulate the motion and compute the cleanliness, i.e., the amount of the solvent deposited at each point on the surfaces. This is done by partitioning the surfaces into triangles using a mesh generation algorithm, and computing the time the center of each triangle stays inside the spray cone during the cleaning motion. This routine is already available in CimStation and IGRIP. If the cleaning analysis shows any points that are inadequately cleaned, the user modifies the cleaning motion using the simulation software. We are currently developing an algorithm that modifies the motion automatically.

3. RULE-BASED PLANNER

Some electronic and mechanical components have

hidden surfaces that need to be cleaned, such as those on electronic circuit boards. When hidden surfaces are involved, the spray cone can never actually reach that area directly, and the desired goal might be stated as: remove all solder flux residue from underneath the components. This problem cannot be treated purely geometrically - solvent flow would need to be modeled and perhaps finite element analysis used to model residue particle dynamics. Numerous assumptions would have to be made about the location and behavior of the residue, and the physical and chemical interactions with the solvent stream. A more fruitful approach is a rule-based specification of the cleaning process.

Currently, a set of rules has been developed for a specific type of printed circuit board, namely surface-mount technology (Figure 2), and the rest of discussion in this section is limited to that case. No single well-defined set of rules appears to exist to describe the spray cleaning of all electronic components. Manual spray cleaning is somewhat haphazard and precise control of the spray motion is difficult to obtain. Therefore, it is difficult to extract a set of rules that will guarantee component cleanliness. This system is intended in part to allow process engineers to interactively describe different path sequences, study their effectiveness, and arrive at a better understanding of the cleaning process specification, so that ultimately the system may be used in a production environment to spray clean electronic and mechanical components.

Rule Specification

Material process engineers have empirically determined the primary spray cleaning rules for surface-mount electronic boards. The part is typically washed, rinsed, and dried, using different nozzles and solutions (heated air, in the case of drying). A number of paths are executed; path parameters such as nozzle speed, nozzle tilt, delay time at a component edge, and exact position of the spray cone are varied depending on parameters such as component size and the proximity of components, as well as the type of cleaning function being performed. The rules loosely expressed by process engineers are formalized, and user-modifiable parameters were identified. The resulting cleaning process specification consists of the following elements:

1. **Cleaning Function:** Current cleaning functions identified are clean, rinse, and dry. Each of these is associated with a specific solution, and a maximum nozzle speed.
2. **Spray Cone Parameters:** In this type of spray cleaning, the nozzles used generate a hollow spray cone, with the importance being placed on the wave front (the irregular ellipse where the spray cone makes contact with a surface). Although this pattern varies with an individual nozzle, the geometry associated with the specific nozzle can be measured.
3. **Component Classification:** Components or chips of different sizes may need to be treated

differently. For example, a path around a component perimeter may need to be repeated twice for a "small" component, and four times for a "large" one. Components may be classified by number of pins or surface area, and the classification given a user-defined identifier.

4. **Path Classes:** Currently six different classes of paths have been identified. These are: mounting board perimeter (follows edge of mounting board), raster (moves back and forth across mounting board), component perimeter (follows edges of each component), final flush (traverses board in one direction), step function (traverses board in one direction, pausing at the edge of each component to flush out remaining particles), and straight line.
5. **Path Templates:** Any number of path templates may be specified for any path class. The user assigns a template identifier, and defines the path parameters. The parameters vary with the path class, but consist of such elements as height of nozzle, nozzle speed, delay time at a component edge, distance from perimeter, etc.
6. **Path Sequence:** The actual path points are generated from the path sequence specified for a given part and the geometric model of the part. The path sequence identifies the spray cleaning functions to perform, the nozzles to use, and a list of the path templates to follow.

4. IMPLEMENTATION

Prototype software has been developed for the geometry-based planner and the rule-based planner, and has been tested with a sample mechanical part and surface-mount electronic board.

4.1 Geometry-Based Motion Planner

We have used the ACIS solid modeler to model mechanical parts mainly because of its open architecture that allows the development of customized geometric codes. We have developed the striping routine using ACIS's Application Procedural Interface (API) routines. Figure 1 shows an F-shaped mechanical part, whose surfaces have been striped. The centerline of each stripe is denoted by a thick line, and they are connected by thick lines to show the ordering of the stripes. Figure 3 shows traces of the last robot link while spraying the vertical surfaces of the part. Note that the spray angle deviates from the surface normals in the concave corners to avoid collisions.

4.2 Rule-Based Motion Planner

A CAD model of the part is created using the Intergraph EMS solid modeler. Software written in the Intergraph PPL programming language extracts the relevant part geometry. The output is an ASCII file listing the number of objects, the number of surfaces in each object, and a set of triangular patches (x, y, z, normal) making up each surface. This file is used to re-create the part geometry for the path planning

software (written in C++), as well as for the robot simulation program (Deneb's IGRIP).

A graphical user interface allows the specification of the cleaning process as defined in the previous section (see Figure 4). Defaults have been provided based on the current set of rules. The user may augment these as needed. A specific path sequence is then specified for a given part to be cleaned. Robot tool path points are generated from the path sequence specification, the spray cone definition and the part geometry. The output is an ASCII file of robot tool path points. Robot commands dealing with nozzles, grippers, etc. are interspersed as required by the process.

The user then previews the paths in a simulated workcell to verify all points are reachable and collision free (see Figure 5). A rule-based cleaning analysis program is being developed that will graphically display the resulting cleanliness of the part in the simulation. If the user is satisfied, the path points and robot control commands are downloaded to the robot, and the actual spray cleaning is performed. Figure 6 shows the actual robot workcell.

5. CONCLUSIONS

We are developing a robotic system that cleans electronic and mechanical components by spraying aqueous solvents. The core of this system consists of the two automatic motion planners that reduce the robot motion programming time from hours to minutes. The geometry-based planner has proven effective for cleaning of exposed surfaces, while the rule-based planner for hidden surfaces has provided the process engineers with a set of tools to rapidly program robot motions through a graphical user interface. The ability to program robot motions for a variety of components in a short time will make robotic aqueous spray cleaning a viable alternative to CFC cleaning and manual aqueous spray cleaning. The consistency and repeatability of the cleaning motion will contribute to the production of high reliability components.

One of the major costs in using robot systems for small batch production is the prolonged motion programming time. This results in low throughput for robotic facilities, which then incurs a large indirect facility amortization cost. Rapid and automatic motion planning algorithms will make robotic systems more cost effective in many areas of manufacturing and environmental restoration.

ACKNOWLEDGEMENT

The authors are grateful to George Bohnert, Allied Signal - Kansas City Division, for his collaboration and assistance in the area of spray cleaning process development and technology, to David Neidig for his work on ACIS codes, and to Michael Hsia for his work on development of the rule-based graphical user interface.

REFERENCES

- [1] Latombe, J.C., *Robot Motion Planning*, Boston/Dordrecht/London: Kluwer Academic Publishers, 1991.
- [2] Hwang, Y.K. and Ahuja, N., "Gross Motion Planning - A Survey," *acm Computing Surveys* vol 24, no 3, pp. 219-292, September 1992.
- [3] Goodman, E.D., and Hoppensteadt, L.T.W., "A Method for Accurate Simulation of Robotic Spray Application Using Empirical Parameterization," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1357-1368, Sacramento, CA, 1991.
- [4] Kieffer, J., "A path following algorithm for manipulator inverse kinematics," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 475-480, Cincinnati, OH, 1990.
- [5] Kircanski, M.V., and Petrovic, T.M., "Inverse kinematic solution for a 7 dof robot with minimal computational complexity and singularity avoidance," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2664-2669, Sacramento, CA, 1991.
- [6] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G. and Shmoys, D.B., *The Traveling Salesman Problem*, New York: John Wiley & Sons, 1985.
- [7] Maciejewski, A.A. and Klein, C.A., "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *International Journal of Robotics Research*, vol. 4, no. 3, pp. 109-117, 1985.
- [8] Suh, S.H., Woo, I.K. and Noh, S.K., "Development of An Automatic Trajectory Planning System (ATPS) for Spray Painting Robots," *Proceedings of IEEE International Conference on Robotics and Automation* pp. 1948-1955, Sacramento, CA, 1991.
- [9] Painting Package, *CimStation User's Manual*, Silma, Inc., 1992.
- [10] Painting Package, *IGRIP User's Manual*, Deneb Robotics, Inc., 1992.

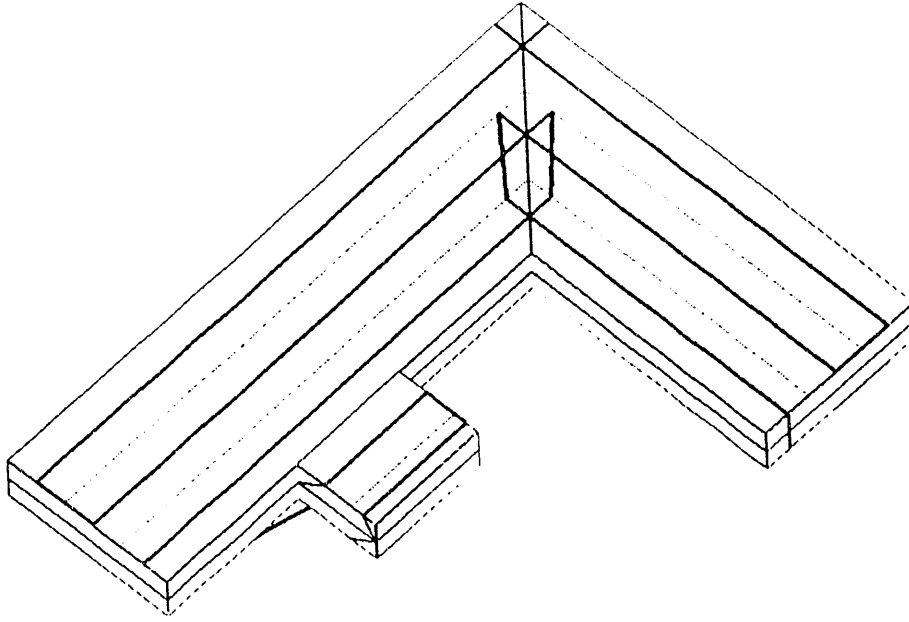


Figure 1. The surfaces of a mechanical component are striped. The path for the center of the spray cone is shown in a thick line. The order of stripes are determined by solving a traveling salesman problem.

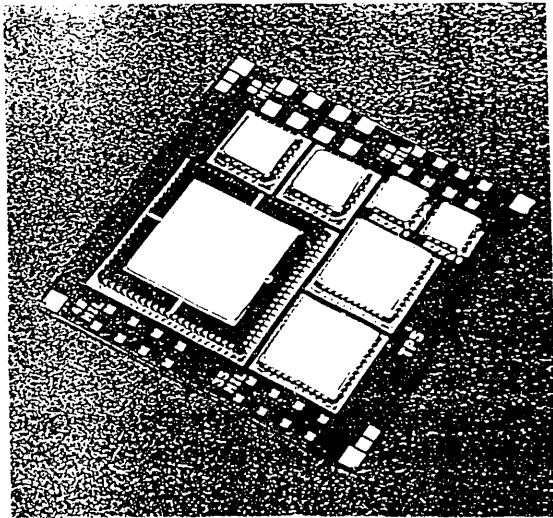


Figure 2. A sample surface-mount electronic board.

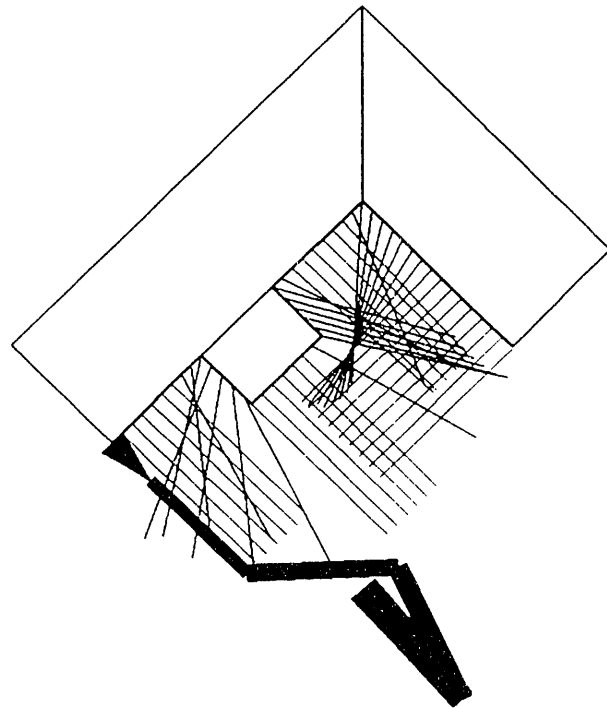


Figure 3. The spraying motion of the robot is executed in simulation. Note that the spray angle deviates from the surface normals in the concave corners to avoid collisions.

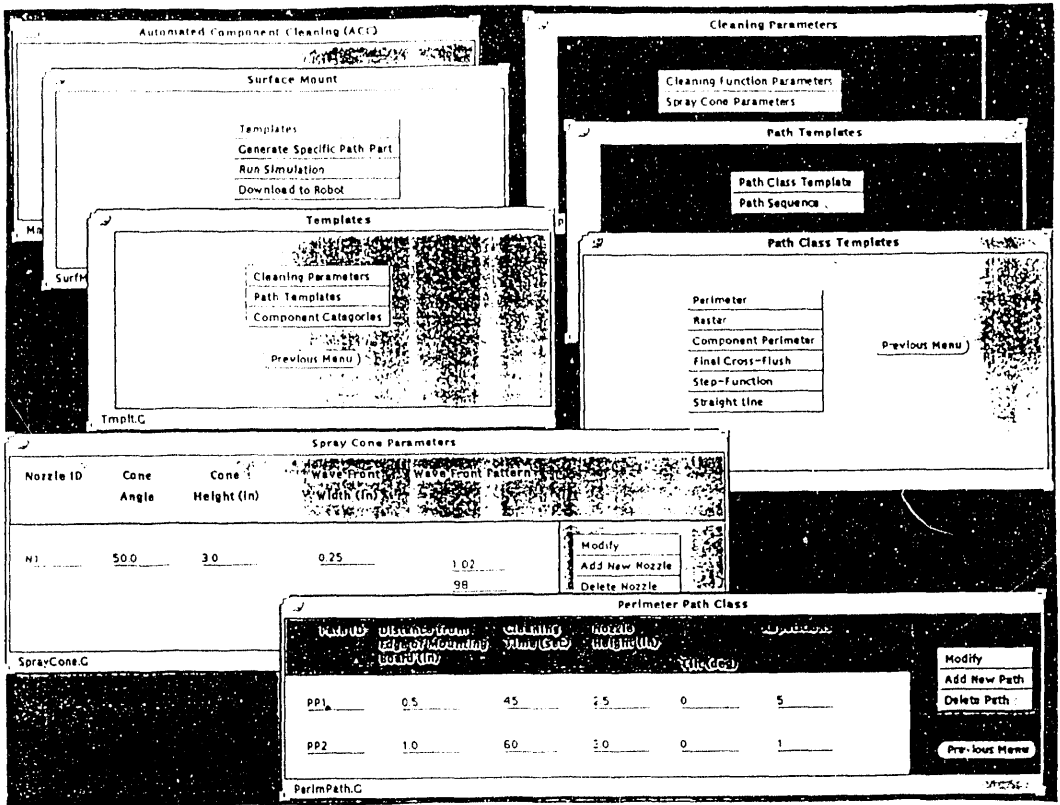


Figure 4. A graphical user interface showing various functions in the rule-based spray motion planner.

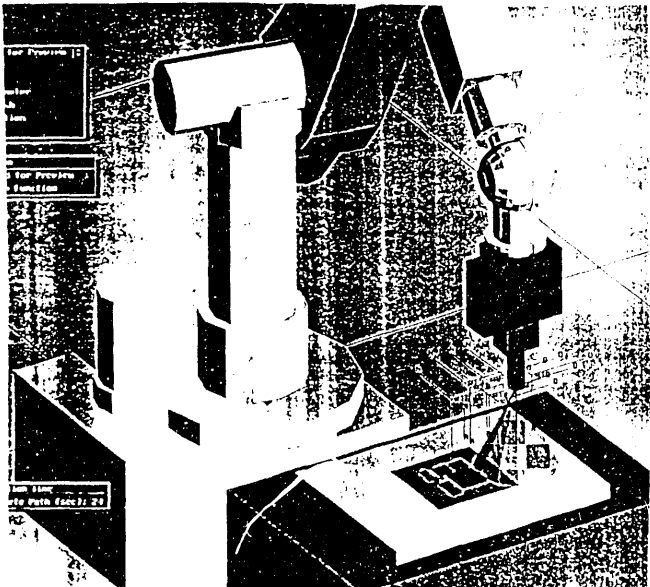


Figure 5. Rule-based motion planning using a simulation software. Cleaning motions are composed with various classes of paths pre-defined in the rule-based planner.

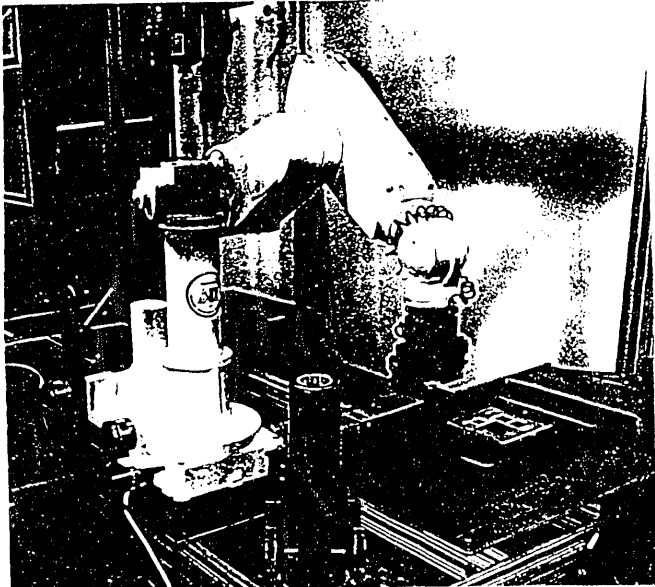


Figure 6. The actual robot workcell for spray cleaning.

**DATE
FILMED**

12 / 14 / 93

END

