SAE TECHNICAL
PAPER SERIES

922404

Integration of a Path Generation Algorithm into Off-line Programming of AIRBUS Panels

L. Graux, P. Millies, P. L. Kociemba, and B. Langlois
AEROSPATIALE

SAE 75 YEARS AEROSPACE

SAE The Engineering Society
For Advancing Mobility
Land Sea Air and Space®
INTERNATIONAL

Aerofast '92
Aerospace Automated Fastening
Conference and Exposition
October 13-15, 1992
Bellevue, Washington

400 COMMONWEALTH DRIVE, WARRENDALE, PA 15096-0001 U.S.A.

Global
Mobility
Database®

All SAE papers, standards, and selected books are abstracted and indexed in the SAE Global Mobility Database.

ISSN 0148-7191
Copyright 1992 Society of Automotive Engineers, Inc.

Positions and opinions advanced in this paper are those of the author(s) and not necessarily those of SAE. The author is solely responsible for the content of the paper. A process is available by which discussions will be printed with the paper if it is published in SAE transactions. For permission to publish this paper in full or in part, contact the SAE Publications Division.

Persons wishing to submit papers to be considered for presentation or publication through SAE should send the manuscript or a 300 word abstract of a proposed manuscript to: Secretary, Engineering Activity Board, SAE.

Printed in USA

90-1203A/PG

922404

Integration of a Path Generation Algorithm into Off-line Programming of AIRBUS Panels

L. Graux, P. Millies, P. L. Kociemba, and B. Langlois
AEROSPATIALE

ABSTRACT

This paper describes a general method for programming robot motions in an industrial environment. The purpose of this study was the off-line programming automation of riveting machines for Airbus (A320-A340) panels and sub-assemblies.

Providing the initial and goal configuration of the robot on the Airbus part, the method determines a collision-free path taking into account CAD models of both the robot and Airbus parts. This method, based on the attractive potential field concept and the random motion technique, has been implemented on a robotic simulation software.

INTRODUCTION

Off-line programming of automated production systems is required for completely integrated CAD/CAM systems (Figure 1). The aim of integrating path generation algorithms is to relieve the programmer of the tasks involved in checking the working trajectories for collisions. The industrial example discussed later describes off-line point-to-point programming for the automated assembly of Airbus panels by rivets. When the large number of points is considered (several thousand), the advantages of such a tool in automatically producing the NC programmes for these panels become obvious.

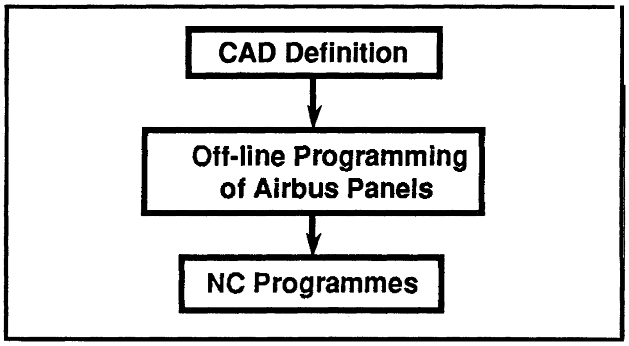

Figure 1. The Intagred CAD/CAM System

THE PATH PLANNING PROBLEM

The problem is to find a path without any collisions between two specified system configurations. The system moves in three-dimensional cartesian space around obstacles described as a set of solids. A system configuration vector is defined as all the parameters required to describe the positions of all points on the moving body. Let M(q) be the set of points in the cartesian space covered by moving bodies for configuration q. A "safe system configuration" is one in which there is neither interference between M(q) and any fixed obstacle nor between any two moving bodies. The term "free space" describes all space available for safe system configurations. In fact, the free space is limited by mechanical stops within the robot. The problem of planning a path can then be expressed as follows: if q_0 and q_f are safe initial and final configurations respectively, calculate a path T(x) [0,1] in free space such that $T(0)=q_0$ and $T(1)=q_f$ (if such a path exists). A local method [1] is used to resolve this problem. Unlike "global" methods [2] [3], local techniques do not use a complete model of the free space. They do not require memory and, at any given time, only use the environment close to the system to modify a setpoint path. These methods are simple but, unfortunately, cannot guarantee getting to the target system since they can lead to a deadlock because of concave arrangements of obstacles. If this happends, it is not possible to extract local data to define a direction of motion which will lead to the target position. In this case, the search continues using random movements. The most informal type of motion is the Brownian movement. The random movements generated are approximations to Brownian movements and are defined as discrete random paths.

THE PATH PLANNING METHOD

This path planner is based on a local method developed by the Stanford Robotics Lab. [4]. It is a potential method in which random movements are generated to resolve situations in which the robot has become jammed.

The algorithms are simple and, consequently, can be quickly implemented. The method is applicable to robots with 6 degrees of freedom or more.

The planning module was first validated on a 6-axis anthropomorphic robot. This helped to resolve many planning problems and gave a more precise view of the characteristics of the various movements. Then the method was applied to an automatic riveting machine with 5 axes plus 2 external axes (URASE).

PRINCIPLE - The path planning module generates three types of movements:

- a "setpoint" motion,
- a "gradient" motion,
- a "random" motion.

The fact that there is no representation of the robot's free space means the planner must drive the robot blindly on. In fact, each new robot position is calculated disregarding the robot's environment. Therefore a collision test must be carried out to validate each new position.

THE SETPOINT MOTION - The setpoint motion (Figure 2) represents the robot control law between the initial position and final one. If the motion is successful (no collisions), the path can be validated directly. If a collision is found, the path is recorded up to the halt point (i.e. the last configuration before collision).

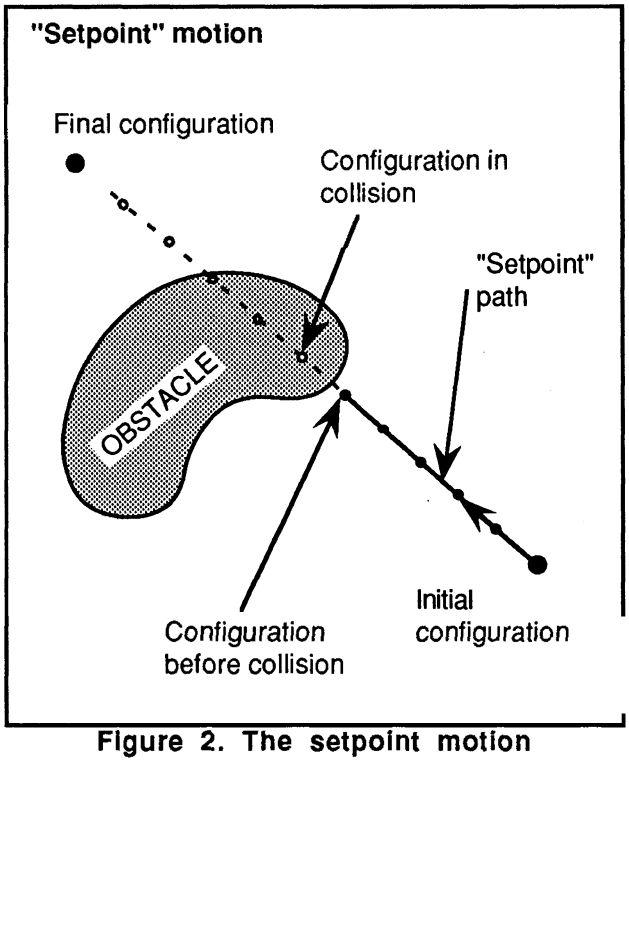

Figure 2. The setpoint motion

A gradient motion is then triggered.

THE GRADIENT MOTION - The gradient motion is used to approach the final position but, simultaneously, bypass the obstacle. It can be performed after a setpoint motion or a random motion. Unlike the setpoint motion, there is no single gradient motion; the purpose is to randomly move the system closer to its final configuration axis by axis.

A gradient motion can terminate in two different situations:

- first case: the gradient motion is successful in bypassing the obstable or in reaching a "saddle point", i.e. a configuration in which the potential is less than that the one of the local minima. Then the path continues with the setpoint motion. (See Figure 3-a)

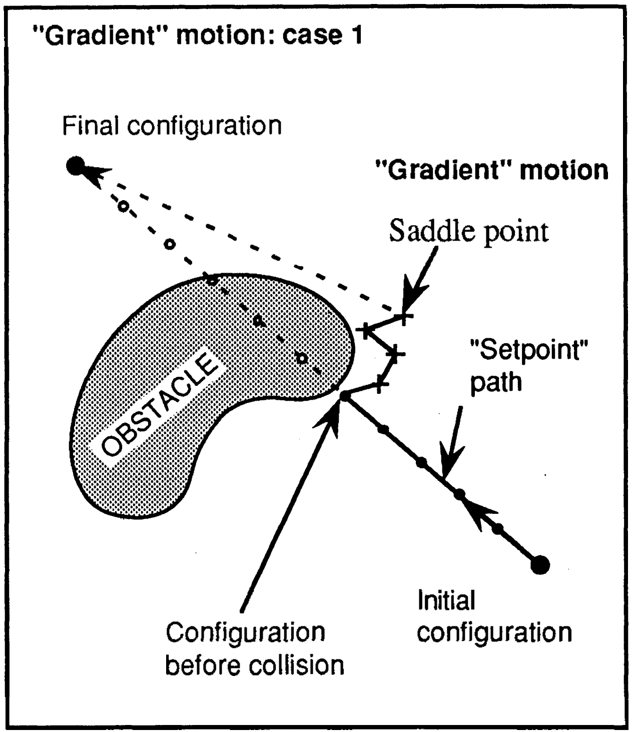

Figure 3-a The Gradient motion

- second case: the gradient motion leads to a local potential minima, i.e. a deadlock configuration. Then a random motion is started. (See Figure 3-b)

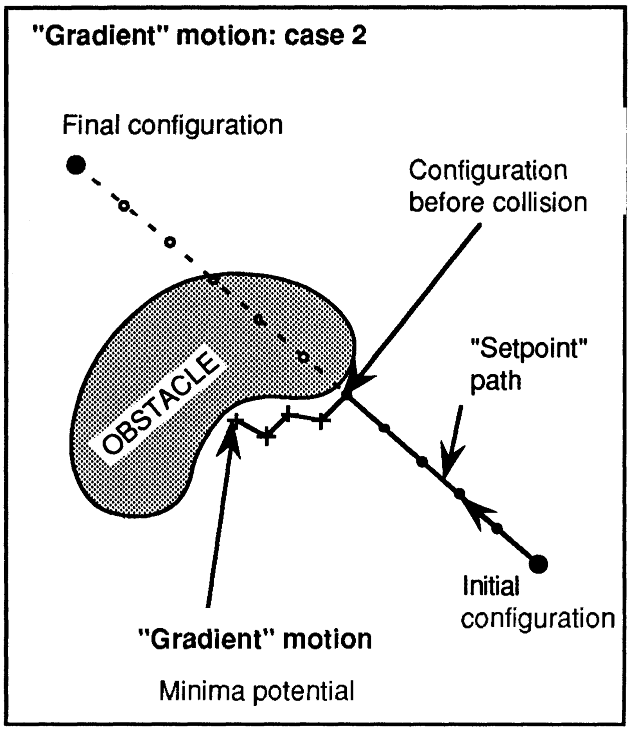

Figure 3-b The Gradient motion

THE RANDOM MOTION - A random motion resolves the deadlock situation by generating a random shift for a given time. (See Figure 4) It is a discrete motion which tends towards a Brownian motion when the discretization increment tends to zero. The duration of the random motion is variable. It starts with a fixed initial value and is incremented at each failure to locate a saddle point.

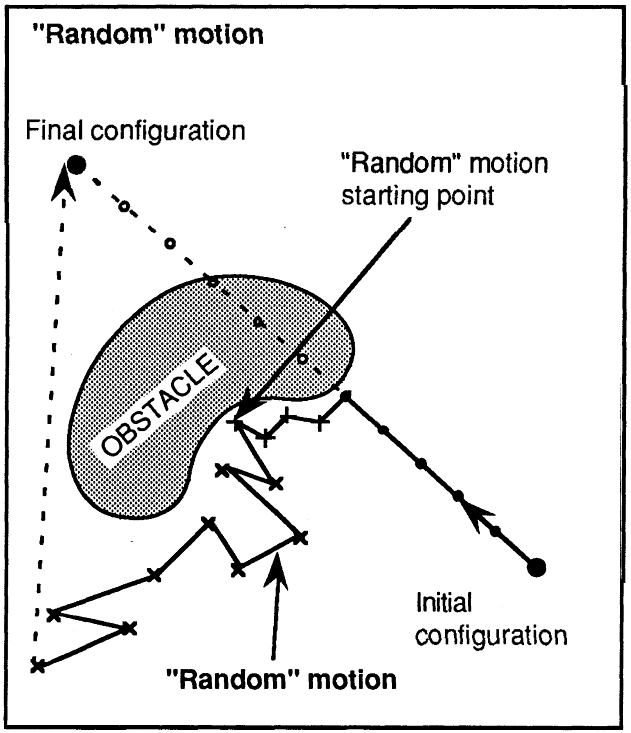

Figure 4. The random motion

Anyway the search stops when a maximum value is reached. The random motion is followed either by a setpoint motion, if a saddle point is found, or otherwise by a gradient motion

SMOOTHING - The last step is to smooth the path. This stage is very important in this method due to the presence of random movements, the trajectories are generally irregular and incoherent; smoothing aims to eliminate all configurations which are not useful to the robot path. Dichotomy techniques or saddle points can be used for smoothing.

APPLICATION OF THE METHOD TO RIVETING PATH PROGRAMMING

ANALYSIS OF THE PROBLEM - Some Airbus panels are automatically assembled on riveting machines. "Assembly" involves riveting structural components (stringers, frames, etc.) to the skin panel (Figure 5). The path planning problem arises when the die must be positioned at the intersection of two structural components. In this case, the machine and die must be moved alternately to generate a path with no collisions. At present, the programmer determines the free path manually. Because the shape, size and location of the structural elements differ from one panel to another, there is a multitude of escape trajectories. Consequently, a general path planning method, able to find all these trajectories, is required.

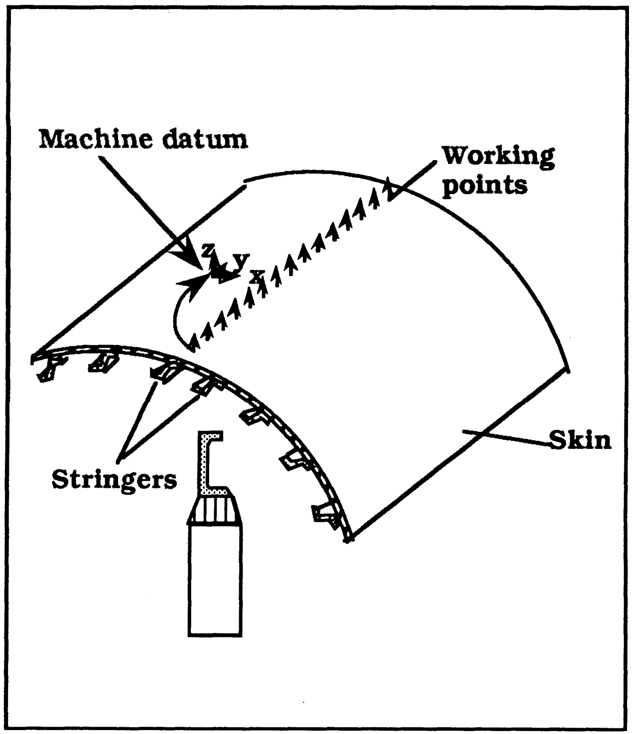

Figure 5. The working points on a Airbus panel

FUNCTIONS AND DATA REQUIRED - The panels are off-line programmed using robot simulation software. Then the following data are required :

* the CAD model of the panel structural elements and the die,

* the datum positions on the panel skin corresponding to the points to be riveted.

The following software functions are also required:

* the direct and reverse kinematics of the machine,

* detection of collisions between CAD entities.

The algorithm section of the programme is separated from the movement management and working environment section. It is important to separate these two types of instruction to ensure portability.

PANEL PROGRAMMING METHOD - A panel is programmed in a fixed sequence (Figure 6). Firstly, the value of the machine axis setpoints is determined for a given working datum and a die position and orientation are validated. A machine configuration is obtained by aligning the defined working datum on the panel with the machine datum which determines the position of the drilling and riveting tools and the head which applies the sealing compound. The z position of the die is determined by placing the die in contact with the component to be assembled. Implicitly, the die is oriented at 90° to the direction of travel but, for difficult positions, this can be changed (See figure 7). A system configuration is therefore defined by the value of the five machine axis setpoints plus the die position and orientation. Then a collision-free path is generated.

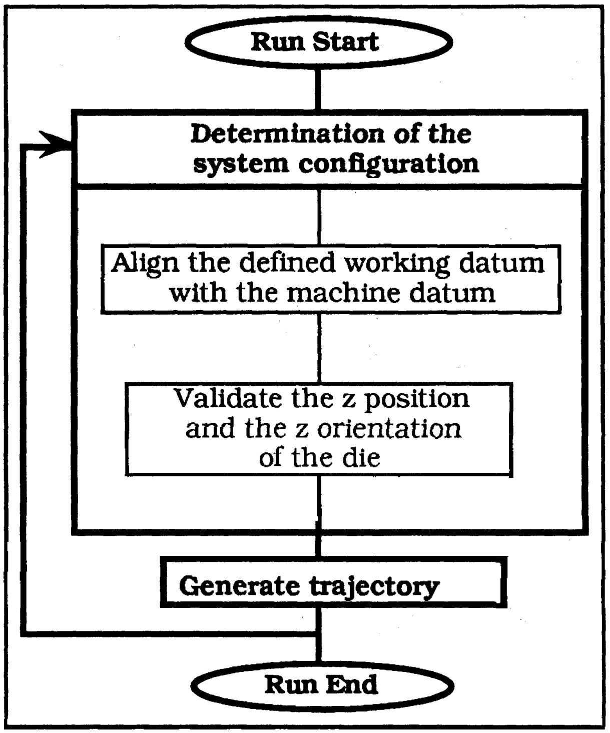
Run Start
Determination of the system configuration
Align the defined working datum with the machine datum
Validate the z position and the z orientation of the die
Generate trajectory
Run End

Figure 6. The programming method

DESCRIPTION OF THE MACHINE KINEMATICS - The kinematics of the automatic riveting machine are complex. The machine has 5 degrees of freedom which is necessary to position the machine datum perpendicular to the position of the point to be riveted. The machine includes a C-frame capable of moving along the x-axis. The tool head is mounted at the top of this frame and a die at the bottom. The die, which clamps the structure component and swages the rivet, is free to move axially and in rotation along z. The panel is mounted on a gantry with two axial degrees of freedom along y and z and 2 rotational degrees of freedom about x and y.

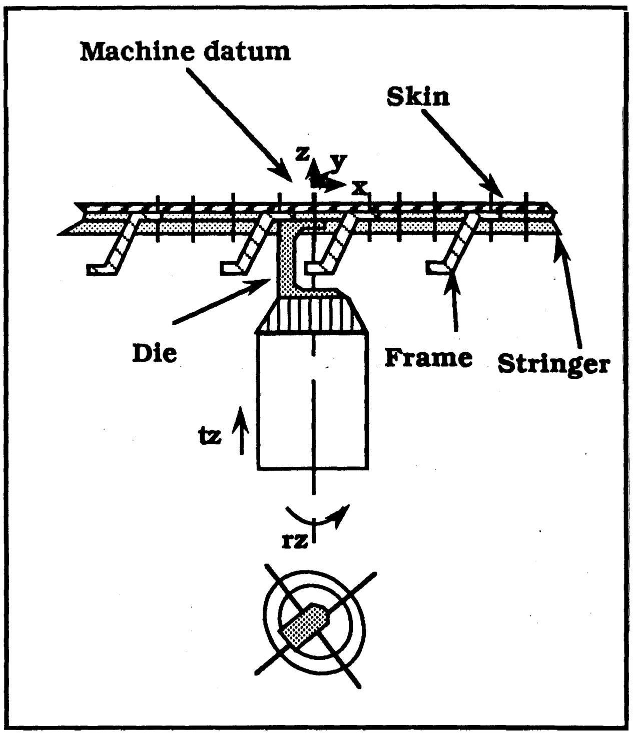

Figure 7. A die configuration

PATH GENERATION - For the riveting machine, heuristic techniques were used to shorten the search time. This was done by limiting the field of action along certain axes during the random motion to avoid searching in sub-spaces where there were known to be no solutions. In this particular case, motion along the z axis and the two panel rotational axes were inhibited and the die motion along the z axis is forced downwards since it is impossible to move over the top of the skin. Another technique is to use the escape trajectories determined by searches at the previous points (Figure 8).

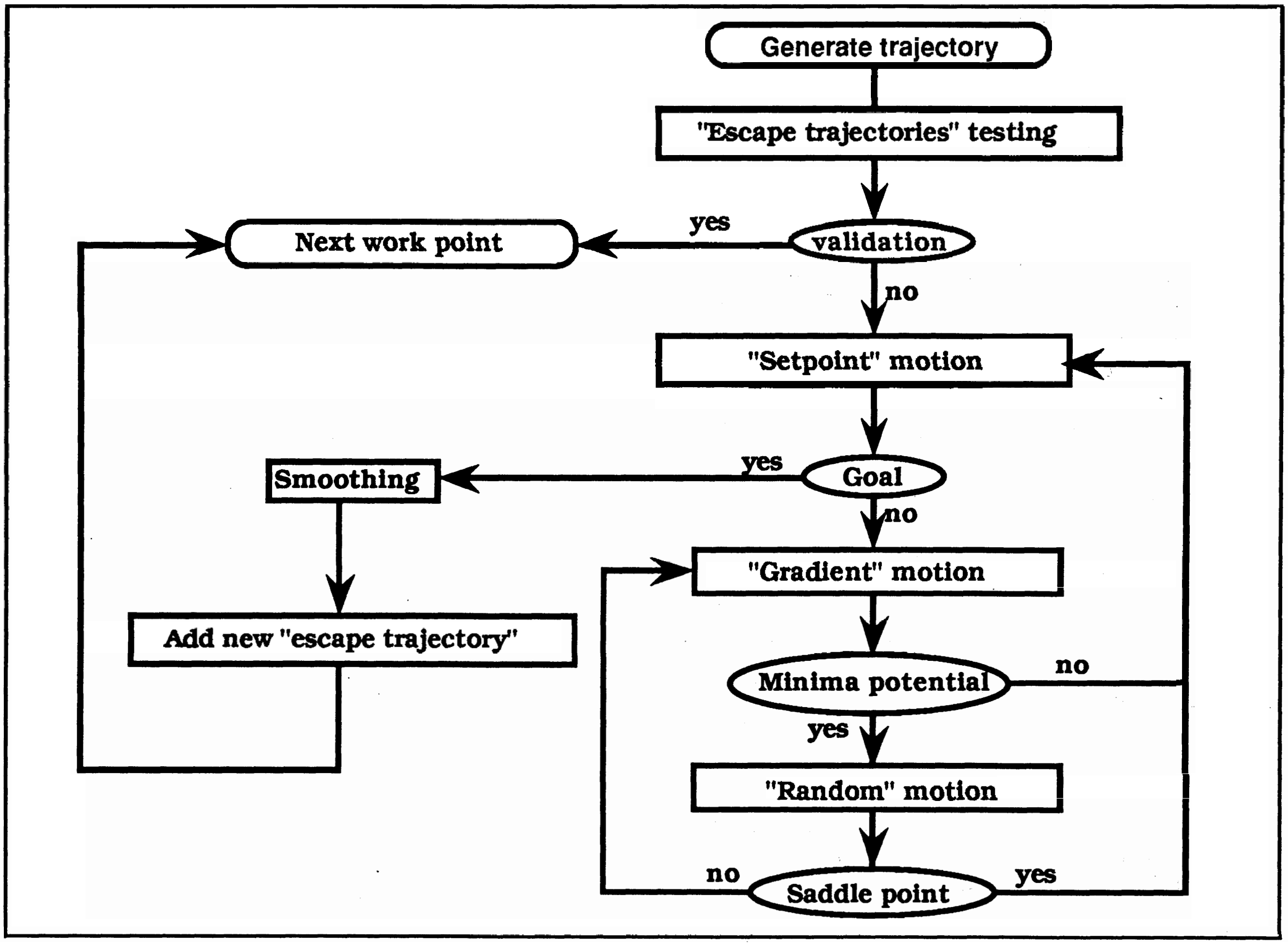

Figure 8. The path generation algorithm

IMPLEMENTATION

This programme was implemented using the CIMSTATION robot simulation software produced by SILMA. SIL (Silma Interactive Language) - a specific language of the software - was used.

In the future it will be possible to implement this algorithm on the ROBCAD simulation software used in the Aircraft Division.

THE RESULTS OBTAINED

Planning with Cimstation takes the operator roughly the same time as was necessary to manually plan trajectories. The path planning module does not save time but can work when the operator is not present and during software slack periods. For reasons of industrial secrecy, we are not allowed to state the time required to plan a complete Airbus panel. The Table 1 displays the results for the generation of a trajectory without heuristics. The Table 2 displays the same example with heuristics. We noticed that the collision detection time is important regarding the computation time.

Table 1
An example of trajectory generation without heuristics

Search Time (sec)	222	236	274	429
Computation Time	7	19	34	36
Detection Time	215	217	240	393
Smoothing Time	23	22	57	48
Configuration Number	85	133	418	448
Setpoint motion	44	62	199	223
Gradient motion	10	26	62	78
Random motion	30	44	156	146
Detection Number	171	211	454	599
Configuration Number after smoothing	8	8	8	8

Table 2
An example of trajectory generation with heuristics

Search Time (sec)	39	47	40	72
Computation Time	2	3	3	4
Detection Time	37	44	37	68
Smoothing Time	20	14	19	18
Configuration Number	65	69	78	84
Setpoint motion	48	48	57	59
Gradient motion	4	8	3	7
Random motion	12	12	17	17
Detection Number	74	81	84	112
Configuration Number after smoothing	8	8	8	8

CONCLUSIONS

The results obtained in planning trajectories for the industrial robot in a complex environment shows that the random motion method can perfectly be applied.

At present, the factors which limit the path planning performance are the time required for smoothing and the large number of anti-collision tests required. These aspects should be improved in the future. Today, the method can deal with all Airbus panels and sub-assemblies and this approach could be extended to high degrees of freedom industrial robots.

REFERENCES

1. O. Khatib, Real-Time Avoidance for Manipulators and Mobile Robots. IEEE Robotics and Automation 1985.

2. B. Faverjon, Obstacle avoidance using an octree in the configuration space of a manipulateur. IEEE Robotics and Automation, Atlanta, Mars 1984.

3. T. Lozano-Pérez, A simple Motion-Planning Algorithme for General Robot Manipulators, IEEE Robotics and Automation 1987.

4. J. Barraquand, J.C. Latombe, Robot Motion Planning : A Distributed Representation Approach., Department of Computer Science, Stanford University, CA 94305, USA.

The automatic riveting machine

URASE