Int J Adv Manuf Technol (2005) 26: 262–267
DOI 10.1007/s00170-003-1728-5

ORIGINAL ARTICLE

S. Mitsi · K.-D. Bouzakis · G. Mansour · D. Sagris
G. Maliaris

Off-line programming of an industrial robot for manufacturing

Received: 13 November 2002 / Accepted: 14 February 2003 / Published online: 10 September 2004

Abstract A procedure for the automatic generation of the robot programming used in manufacturing operations is introduced in the present paper. The off-line programming system developed here includes graphical simulation of the robot and its workcell, kinematic model of the robot, motion planning and creation of the NC code for manufacturing process. The proposed system is applied in a robot with five revolute joints for manufacturing operations and in a robot with six revolute joints for welding operations.

Keywords Robot · Off-line programming · Simulation · Welding

1 Introduction

Robotics applications in manufacturing lead to the reduction of production time and the improvement of the quality of the workpieces. Small and medium enterprises that produce a wide variety of products require a method that generates the NC code for processes automatically. The robot off-line programming using a CAD system has the potential to produce a visual presentation of the robot when performing its task and to eliminate in the planning stage problems of robot reach, accessibility, collision, timing, etc. In this way significant time economy can be achieved, because the actual manufacturing would only have to be interrupted briefly, while the new programs are downloaded into the workcell control computers. The robot off-line programming using a CAD system has been the subject of much research in recent years. Computer graphics simulation of the robot and its workcell can be realized with different models such as wire-frame and solid models [1]. These models and adequate algorithms can be used for collision detection and for kinematic and dynamic behaviour of the robot [2,3,4].

In the present paper, the procedure for automatic generation of the programming of the manipulator used for manufacturing is described. The off-line programming system developed here includes graphical simulation of the robot and its workcell, kinematics of the robot, motion planning and creation of the NC code. With the aid of this procedure it is possible to simulate the behaviour of the robot during the process. The developed system is demonstrated in two cases: a robot with five degrees of freedom and five revolute joints (RV-M1) used for general manufacturing processes and a robot with six degrees of freedom and six revolute joints (RV6) used for welding.

2 Graphical simulation

In the first step of the procedure, the graphical simulation of the manipulator, tool, workpiece and workspace is achieved using the SolidWorks program [5]. Geometrical characteristics such as size and shape of the robot links and the kinematics are introduced into a database. The description of the robot kinematics contains information about the degrees of freedom, the type of joints, the Denavit-Hartenberg parameters [6], the joint functionality limits, etc. After the drawing of the manipulator links in solid three-dimensional space, they are assembled taking into account the position of the joints and the relative motion between the links. Figure 1 shows the simulating model of a robot with five revolute joints.

The simulating model of the robot environment contains geometrical data of the workspace including the workpieces, which will be processed and can be

S. Mitsi (✉) · K.-D. Bouzakis · G. Mansour
D. Sagris · G. Maliaris
Laboratory for Machine Tools and Manufacturing Engineering,
Mechanical Engineering Department,
Aristoteles University of Thessaloniki,
54124 Thessaloniki, Greece
E-mail: mitsi@eng.auth.gr
Tel.: +30-2310-996043
Fax: +30-2310-996059

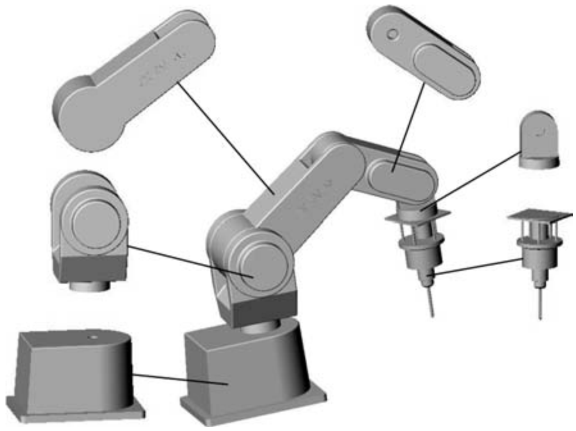


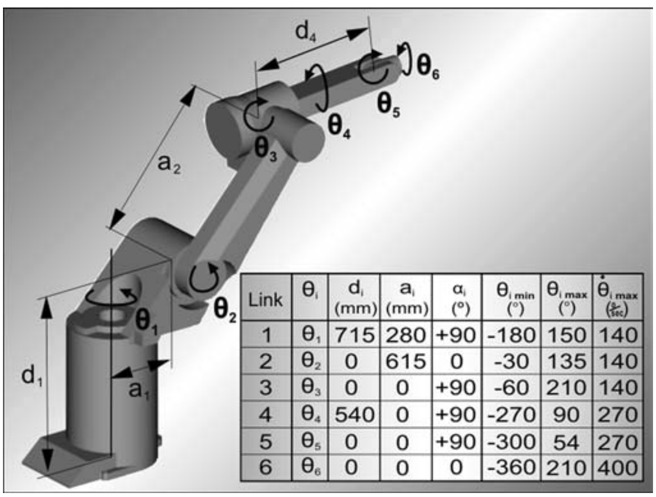
Fig. 1 Robot with five revolute joints

described in the same way as the geometry of the robot.

3 Robot kinematics

In order to plan and simulate the robot motion, the forward and inverse kinematics must be solved. In this view the robot consists of an open spatial kinematic chain. Figure 2 shows the simulating model of a robot with six revolute joints. Using the homogeneous transformation matrices and the Denavit-Hartenberg parameters given in the table of Fig. 2, the motion equations of the manipulator are developed. These equations can be used for the direct or inverse kinematics of the manipulator.

For a prescribed position and orientation of the end-effector, the problem of the inverse kinematics of the manipulator is solved analytically. The manipulator in Fig. 2 has a closed form solution, because the axes 4, 5 and 6 intersect [7]. In this way all eight solutions of the



Link	θ_i	d_i (mm)	a_i (mm)	α_i (°)	$\theta_{i\,min}$ (°)	$\theta_{i\,max}$ (°)	$\dot{\theta}_{i\,max}$ (°/sec)
1	θ_1	715	280	+90	-180	150	140
2	θ_2	0	615	0	-30	135	140
3	θ_3	0	0	+90	-60	210	140
4	θ_4	540	0	+90	-270	90	270
5	θ_5	0	0	+90	-300	54	270
6	θ_6	0	0	0	-360	210	400

Fig. 2 Robot with six degrees of freedom and its Denavit-Hartenberg parameters

sets of the joint variables and the corresponding configurations of the robot are determined [8]. From the sets of the joint variables the one that guarantees collision avoidance, joint limits and avoidance of singular configurations is chosen. The CAD package used has a high potential for collision detection. The joint angles are calculated for every point along the end-effector path using the Fortran programming language.

4 Motion planner

From the workpiece CAD model, the programmer generates a list of the initial and final points of each movement and the workpiece features to be processed as position, material, width, depth of the process, etc. The motion planner, based on the robot and workpiece CAD model, is divided in gross and fine motion planning.

The gross motion planner deals with the planning of the approach and departure motions of the end-effector to and from the process operations. The generation of a collision-free path is realized with a ‘‘hypothesize and test’’ approach [9]. The approach utilizes heuristic rules to provide an efficient solution. For a candidate path from the start to the goal end-effector pose, the user can define the number of points, and using a linear or spline interpolation the intermediate points are calculated. At each location, the inverse kinematics is solved, and considering joint limits the collision avoidance is checked. If a collision is found, a new candidate path is proposed by examining the obstacles involved in the collision. The procedure continues until a collision-free path is found. An example of a collision-free path generated by the gross motion planner is shown in Fig. 3.

The fine motion planner plans the operation path from the initial to the final point of each process. During the process the robot must hold the tool at the correct orientation, at the correct distance and move at a constant velocity. Taking into account the tool holder orientation, the orientation of the end-effector relative to the base frame of the robot is determined. Due to the fact that sometimes such orientations lead to a collision of the end-effector with the workpiece, the user can modify the end-effector orientation angles until the collision is avoided. A linear interpolation is used for trajectory planning.

The obtained trajectory is saved to file and the graphical simulator displays the sequence of robot motions for verification before the actual process.

5 NC-code generation

To generate the NC code, the robot motion program is completed with commands regarding the process (pulse velocity, spin velocity, coolant flow, etc for manufacturing processes and arc current, arc voltage, welding speed, feed wire, etc for welding processes), taking into account the technological data of the process (workpiece

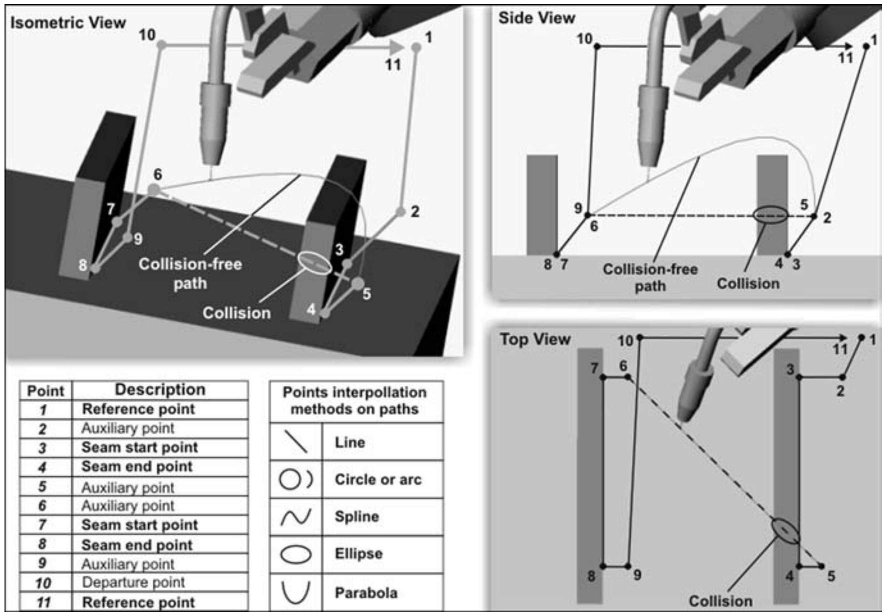


Fig. 3 Path optimising for obstacles avoidance

and electrode material, process position, etc) and the interpolation facilities of the robot controller. The program can then be downloaded to the robot controller.

The structure of the NC-code generating system for manufacturing processes is illustrated in Fig. 4. The path planning part, developed in the Visual Basic environment, simulates manipulator movement working parallel with the inverse kinematics problem solution, which is solved in Fortran and parallel with the CAD models constructed in SolidWorks design software. While taking into account the process parameters, path planning simulation-software creates the NC code for manufacturing automatically.

Simulation software is developed so that the method of automatic trajectories programming mentioned above is easy for the user [10]. All the movements set on the platform of this software are interactive with the SolidWorks platform to ensure user supervision of trajectory programming. The movement control of the robot is achieved through the end-effector position and orientation definition. In order to improve the accuracy of the process simulation, a procedure which interpolates points along the selected trajectory is carried out. This linear interpolation refers not only to the coordinates but also to the orientation of the end-effector. The trajectory is saved to file, in order to have easy access to each trajectory.

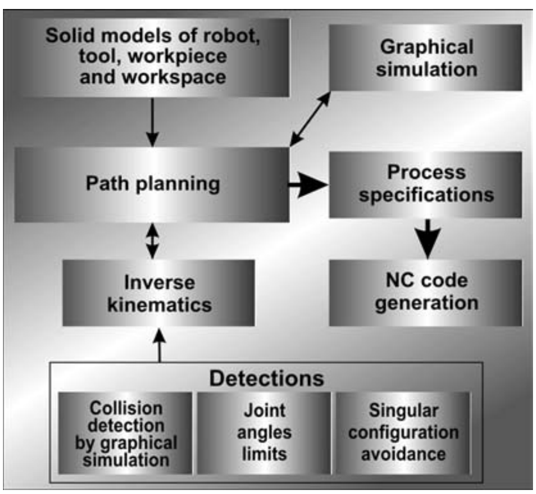


Fig. 4 Automated robotic welding system structure

The developed software includes three forms. The main form called ‘‘Points Base’’ is presented in Fig. 5 and contains the points along the trajectory. The end-effector trajectory can be produced in four ways:

– By choosing a pre-designed path
– By choosing a designed path of lines and splines in SolidWorks
– By setting points, which belong to the trajectory of the end-effector, in the 3D workspace of the robot
– By choosing vertices or edges (lines, arcs, splines, parabolas and ellipses) of the part to be processed

The path points are presented in a table and the parameters of the selected point can be changed. For each point, there is the ability to set the coordinates (X,Y,Z) and the orientation angles (A,B,C) of the end-effector to approach the target point with the right orientation, according to the process techniques. The joint angles values are presented for each point, and there is the ability to see the overstepping of the joint angles limits and the collision of any part of the robot with an obstacle in the workspace. The ‘‘welding’’ and ‘‘OSAP’’ boxes can be marked to set the starting point of a process (welding in the present example) and an ‘‘Oscillation Auxiliary Point’’ respectively. There is also the opportunity to interpolate points among the final trajectory points, to simulate the entire trajectory

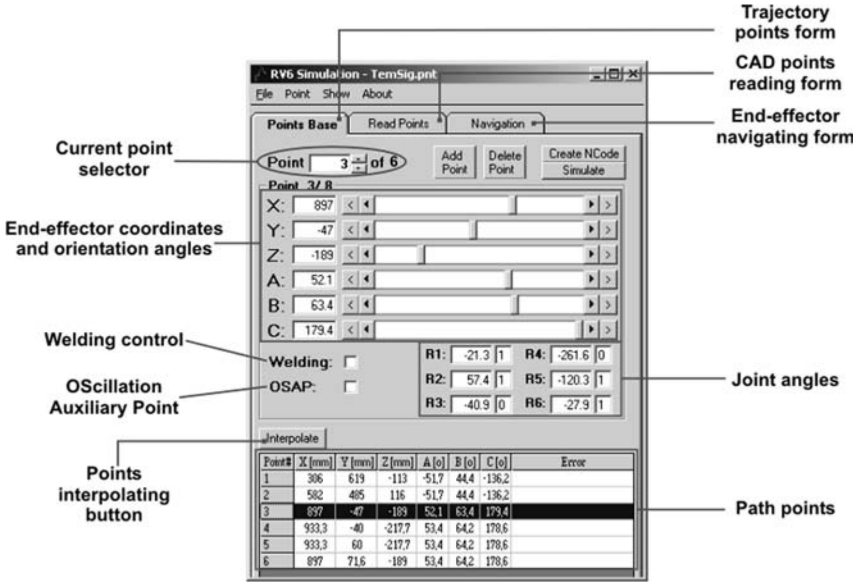


Fig. 5 Robot control program

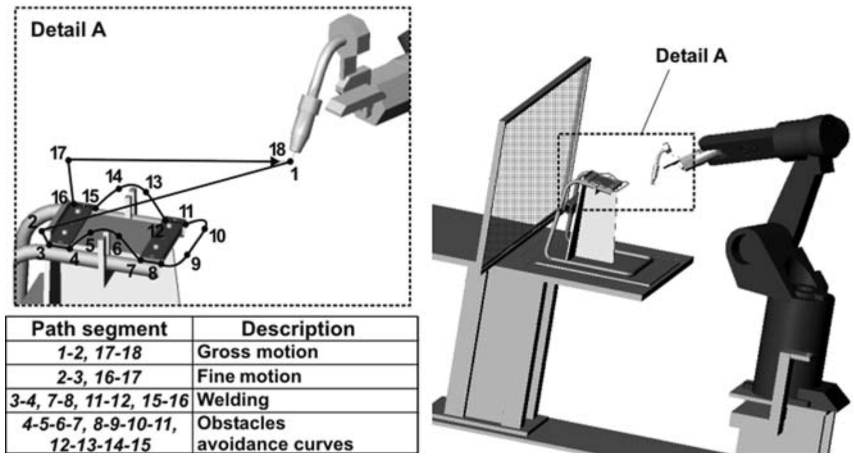


Fig. 6 Graphical simulation of robot and its workcell

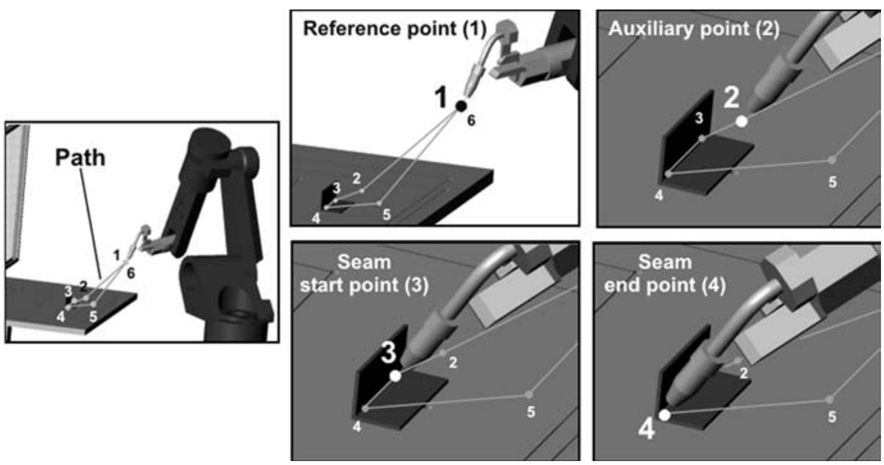


Fig. 7 Four points of the welding process path

and check the point's parameters in the table of points.

The second form called "Read Points" is the channel connecting the user with the environment of SolidWorks software. All the selections of the workpiece (points, curves, surfaces, etc) in the SolidWorks space can be read, identified and checked for sequences among them. An adjustable number of interpolated points are

Table 1 Welding process parameters

Parameter	Value	Units	Specifications
Welding method			Metal Inert Gas (MIG)
Current intensity	120	A	Direct
Arc voltage	21	V	
Wire feed	7,6	m/min	
Welding velocity	3	mm/s	
Gas flow	20	l/min	Argon
Wire diameter	0,8	mm	Material: Steel alloy
Wire extrusion length	15	mm	
Sheet thickness	4	mm	Material: Steel alloy
Welding length	100	mm	
Electrode orientation angles			

calculated for each selection. All these points (selected and interpolated) can be added to the points of the final trajectory.

The third form called ''Navigation'' is used for the control of the end-effector's coordinates and orientation, without changing any point of the trajectory. Hypothesizing a position, it checks the overstepping of the joint angles limits and the collision of any part of the robot with an obstacle in the workspace. Passing these tests, the programmed point could be added in the points base trajectory.

NC code for the robot control is created automatically, working with the three forms on parallel, hypothesizing and testing points, checking the joint angles limits and the collision-free movement, simulating the movement and reaching the desired result of trajectory.

6 Application

The above methodology is applied in a six degrees of freedom manipulator used for the welding of office furniture frames [11]. An example of the graphical simulation of the robot and an office chair base adjusted on its worktable is presented in Fig. 6. The end-effector path is shown in the figure also (Detail A). Four linear segments 3–4, 7–8, 11–12 and 15–16 for welding and three intermediate spline curves 4–5–6–7, 8–9–10–11 and 12–13–14–15 for movement with collision avoidance compose the path. The sequence of the welding lines is picked in order to reduce the welding process duration.

Another application of the proposed methodology is presented in the following. The graphical simulation of the robot and workpieces to be welded is shown in Fig. 7. The programmed points of the entire path are inserted in point's table in Fig. 5. The inverse kinematics problem is solved for all points along the trajectory. Four main points of the welding process are illustrated in Fig. 6 in the SolidWorks environment.

Welding parameters used in this application like current intensity, arc voltage, wire feed, etc presented in Table 1, are selected so as to satisfy all welding technology conditions. These parameters are determined in the

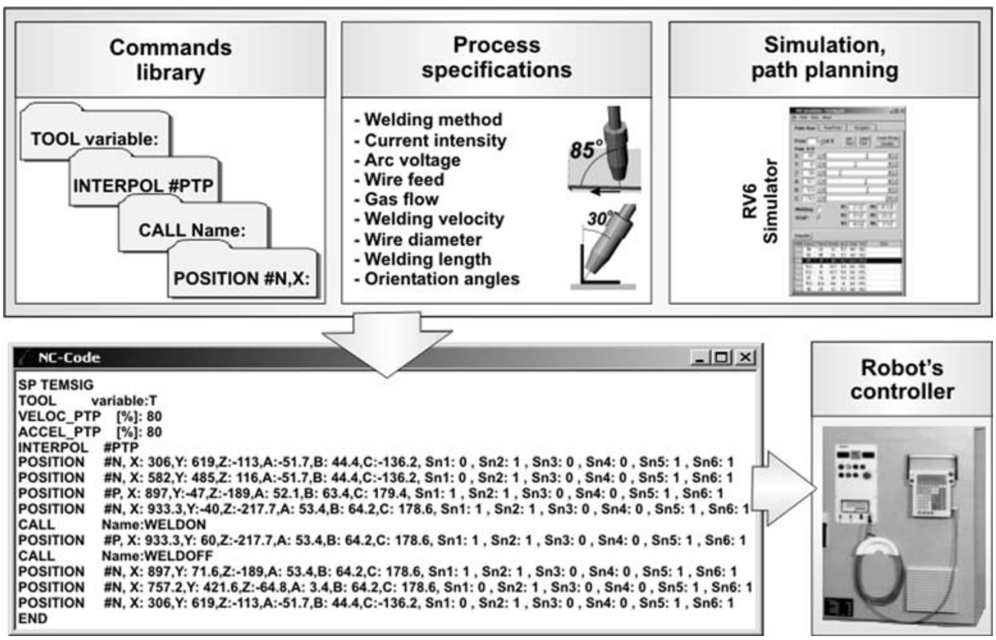
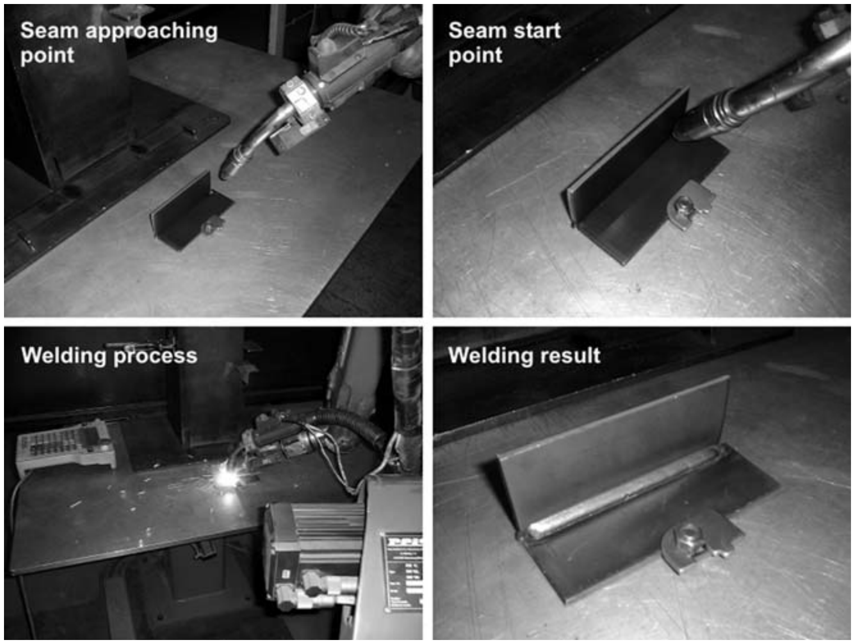


Fig. 8 NC code for welding

Fig. 9 Application in robot real workspace



procedure "WELDON" called by the main program of NC code. The "WELDON" procedure uses linear interpolation movement (CP_Line) and constant velocity.

The NC code presented in Fig. 8 is created using the programmed points of the entire path. Applying it to the robot controller, it executes this code, moves the end-effector to the specified positions and generates the seam as presented in Fig. 9.

7 Conclusions

By means of the developed procedure the NC code for processes can be generated, taking into account the technological data of the process and the interpolation facilities of the robot controller. The proposed system can be further used for optimisation of the process planning of workpieces by reducing the cycle duration of the workcell.

The off-line programming application using a CAD/CAM system significantly reduces the robot downtime and the production process becomes more efficient.

Future development of the robot's dynamic model is required in order to improve the path planning of the end-effector and reduce limits to the control of movements such as speed, acceleration and actuator torque.

Acknowledgements Special thanks are expressed to Dromeas industry for the opportunity to perform experiments on their robot.

References

1. Megahed SM (1993) Principles of robot modeling and simulation. Wiley, New York
2. Mitsi S, Bouzakis K-D, Mansour G (1995) Optimization of robot links motion in inverse kinematics solution considering collision avoidance and joint limits. J Mech Mach Theory 30:653–663
3. Mitsi S, Bouzakis K-D, Tsiafis I, Mansour G (1997) Dynamic behavior simulation of a manipulator with five degree of freedom considering joints friction. J Balkan Tribological Assoc 3:51–60
4. Mitsi S, Bouzakis K-D, Mansour G (2000) Automatic NC code generation for welding by means of an industrial robot. In: TCMM, Technical Publishing House, Bucharest, 40:311–316
5. SolidWorks (2001) User's guide. SolidWorks Corporation
6. Denavit J, Hartenberg RS (1955) A kinematic notation for lower pair mechanisms based on matrices. ASME J Appl Mech 22:215–221
7. Craig JJ (1986) Introduction to robotics, mechanics and control. Addison-Wesley, Reading
8. Mitsi S, Bouzakis K-D, Mansour G (1999) Automatic NC code generation for welding by means of an industrial robot considering workcell. In: Proc 5th EEΔM Conf Mach Tools Manuf Process, Thessaloniki, pp 367–375
9. McKerrow PJ (1993) Introduction to robotics. Addison-Wesley, Wokingham
10. Sagris D (2001) Software development for off-line programming of industrial robot RV6. Dissertation, EEDM, Aristoteles University of Thessaloniki
11. Reis Robot Manual. Reis GmbH and Co Maschinenfabrik