Tomasz KOCH, Paweł KOWALCZEWSKI, Arkadiusz SZYPICYN,
Andrzej KOCEŁUCH, Marek DUTKOWIAK

The Institute of Mechanical Engineering and Automation
The Technical University of Wrocław, Poland

AUTOMATED ROBOT PROGRAM GENERATION USING THE SIMULATION SYSTEM FOR ROBOTISED WORKCELL

Summary. The paper presents a concept of computer-assisted robot programming within RME (Robot Modelling Extension) - the robotised workcell simulation system for personal computer. Integrated versus non-integrated methods of part programming based on CAD data are compared. The integrated system for modelling, programming and simulating robotised workcell is described as a tool for automated obtaining the robot program in BAPS - the Bosch language for robots - from CAD drawings. An example is given for the DFG robot performing the gluing operation of the car body window.

1. CAD/CAM concept

Since many years computer technologies have been developed leading to broad computer applications in manufacturing environment. The computer based technologies known as CAx techniques (CAD, CAM, CAPP, CAQA, etc.) are growing rapidly and start to influence each other. These interactions lead to integration of existing subsystems into one CIM system, making many processes involved in planning, design and manufacture more economical and convenient than they used to be.

1.1. Traditional part programming

NC part programming is the process of obtaining - from the detail drawing of a part - control data, a program which can be used by a numerically controlled device to handle (i.e. to produce or manipulate) the part. Traditional part programming takes place when programs for numerically controlled machines (e.g. machine tools, robots) are created within non integrated environment. The main subsequent steps for developing the NC program are presented in Figure 1. Firstly, having a concept of the part to be implemented, a part drawing must be drawn using the CAD system. Then, the finished drawing is stored in a local CAD database. To create the NC program for a specific part, the programming specialist must extract proper data from the engineering drawing and create the NC program manually. It is a very time-consuming process, therefore integrated computer aided techniques have been developed.

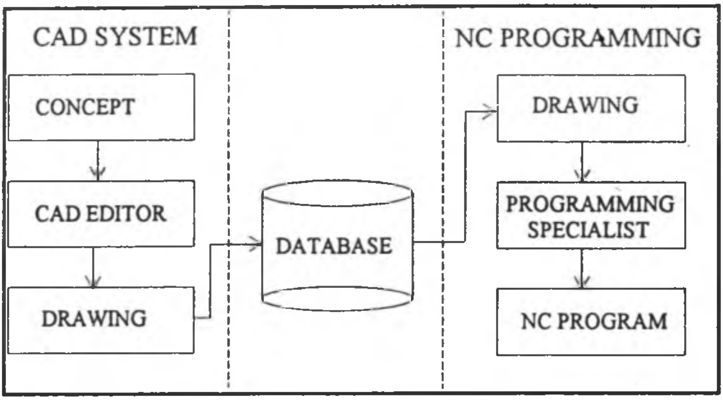

Fig. 1. Traditional NC part programming

1.2. Part programming based on CAD data

The CAD/CAM techniques allow all processes involved in design and implementation of a new product to be made faster and more efficient. Within integrated systems a control program for particular machine can be generated directly from CAD descriptions of the part. This is what is usually known as CAD/CAM - the link between computer aided design and computer aided manufacture. By capturing the engineering drawing directly from the CAD system there is already a great deal of data in the computer representing the form of the product. It is more effective to use this data for all tasks requiring product data than generate this data again or extracting it manually for each specific task.

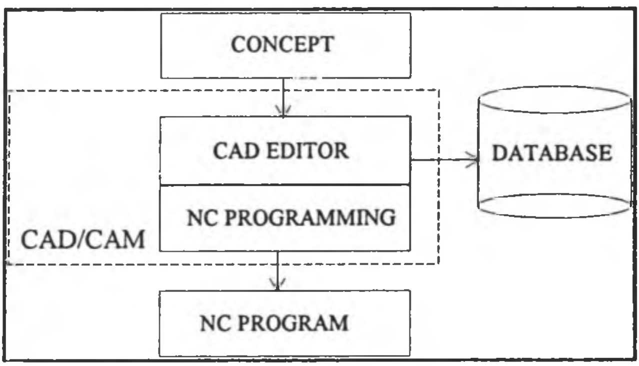

Fig. 2. Part programming integrated with CAD system

As one would expect, it would be a great step forward if the explicit robot program for motion control of the particular robot tasks could be generated directly from CAD description using the same philosophy as NC programming. The robot programs should be generated using robot modelling systems integrated with the simulation system based on graphics software to ensure that the tool motion generated by CAD/CAM module is possible and collision free. Such graphics software systems have been developed in the world since a few years leading to better utilisation of available hardware. One of the CAD oriented tool for implicit and explicit definition of robot trajectories was developed in Institut für Prozeßrechnentechnik und Robotic at Universität Karlsruhe, Germany [4].

2. CAD/CAM within RME System

2.1. RME simulation system

The increasing costs of programming within all costs involved in manufacturing may question economical effectiveness of time-consuming programming methods such as play-back, teach-in or on-line textual programming. These methods have several important disadvantages:

- all production line must be stopped when any machine needs to be reprogrammed, and
- complicated programming and testing process involve high-skilled personnel for a long time.

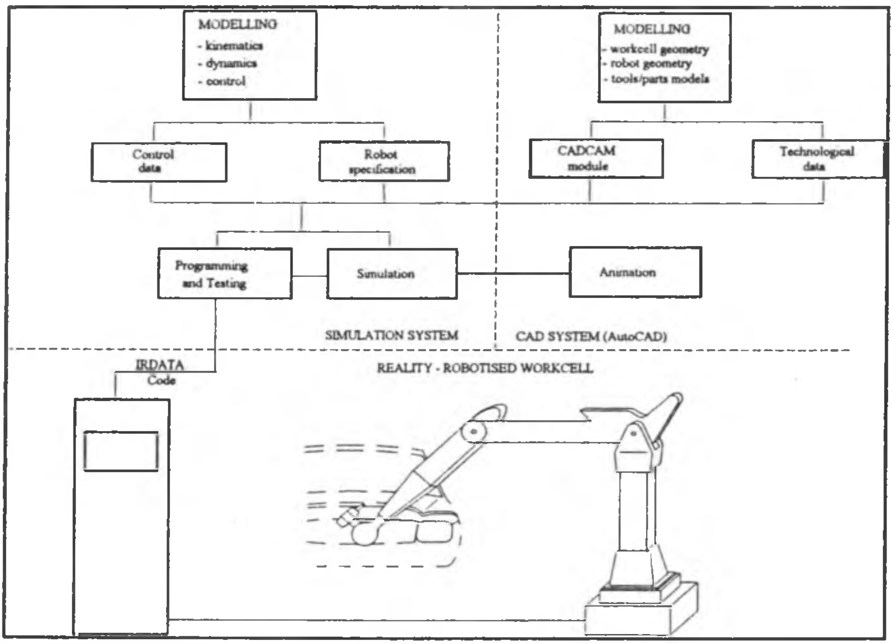

Fig. 3. The block structure of the RME System

A new concept of programming of robotised workcells has been realised within RME System (Robot Modelling Extension) for modelling, programming and simulation of robotised workcells [1][2] (fig. 3.). The System allows users to:

- model robot kinematics, dynamics and control system,
- model robot and workcell geometry

- off-line robot's programming
- simulation of programs

The RME System works within CAD environment. It is written in C language as an ADS (the AutoCAD Development System) application for AutoCAD system. The integration with the well-known CAD package for IBM PC and other computer platforms provides many important advantages, including a plentiful set of ready-made graphics tools (e.g. 3D solid modelling, animation of models, collision control between robots, machines and other objects).

Modelling of robot's kinematics uses data obtained from compilation of an RDL (Robot Description Language [1][3]) file, and modelling of a robot's geometry uses an AutoCAD AME (Advanced Modelling Extension) module.

A workcell is designed with a model of a robot taken from a database. The model is positioned in the workcell's base coordinate system. The complete workcell model may be stored in another database. One of the modelled workcell is presented in fig. 4.

Having created the model of the workcell the user may prepare robots' programs to be simulated in the system.

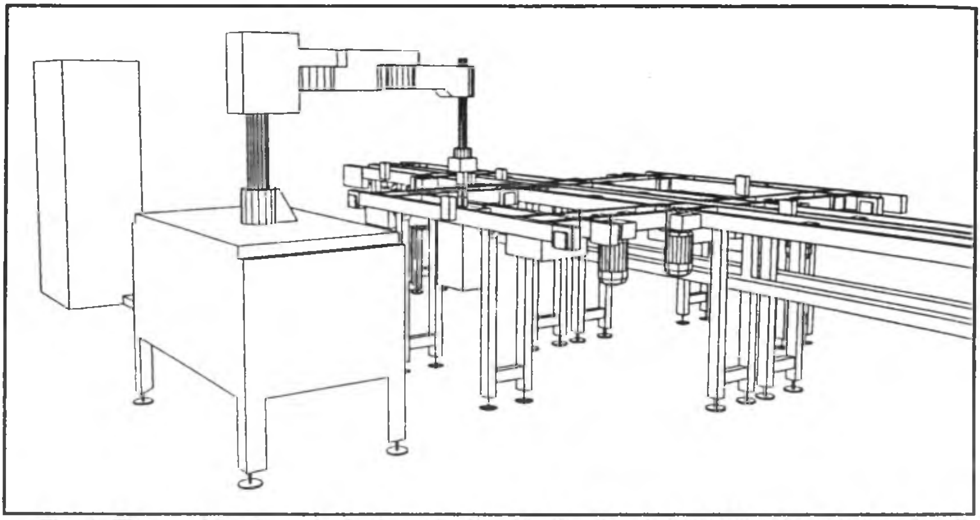

Fig. 4. The model of a robotised workcell with the Bosch SCARA 800 robot and pallet transportation system

2.2. CAD/CAM module within RME

To achieve the full CAD/CAM integration within RME System a new module has been developed. That module generates automatically the robot program basing on an existing trajectory obtained from a CAD system. The trajectory can be defined either outside or inside the simulation system using the AutoCAD package. The form of the trajectory can be just drawn as a line or extracted from the already designed parts. It can be a contour of one or more parts, a section line of a part or intersection line of some parts. The only requirement for lines so obtained is to be a two- or three-dimensional polyline - one or more. For that reason, the CAD/CAM system is well suited for line and surface dependent applications like arc welding, painting, spraying or gluing operations. Fig. 5 shows a polyline extracted as the intersection of two elements to be joined by an arc welding robot.

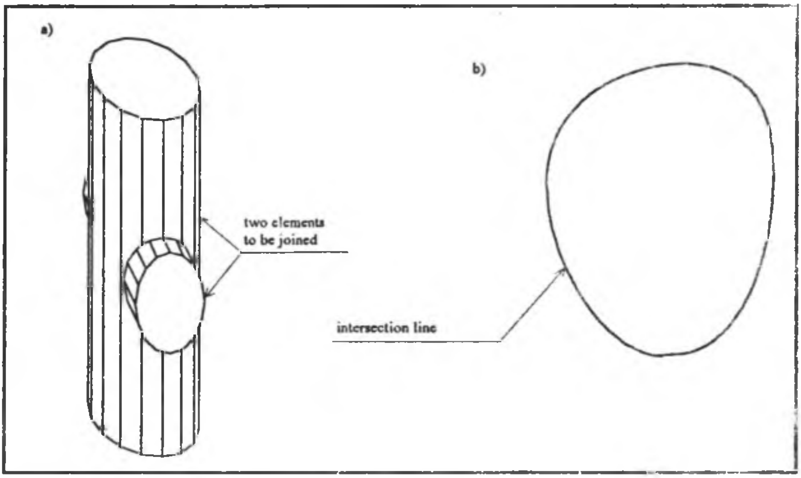

Fig. 5. Obtaining the trajectory from the part drawing:
a) two elements to be joined along the intersection line,
b) the intersection line (enlarged)

Beginning work with the program generator, firstly the system has to be informed by the user how the end-effector orientation is described in a particular robot (e.g. which Euler angles the robot uses). Then, one can associate additional data with any vertices of the polyline using the AutoCAD graphics editor. It includes the following parameters:

- time, velocity or acceleration of the robot's effector motion between the previous and the current vertex of the polyline,
- delay time at that point of the trajectory,
- the effector state at this point (open or close the gripper, spray on/off, etc.)
- the end-effector orientation.

All these parameters are treated as extended entity data associated with the proper vertex of the polyline and are stored in a disc file along with the drawing of the polyline. These parameters must be defined by the user. To effectively control the orientation of the end-effector at the proper vertex of the polyline, the system generate auxiliary arrows oriented according to the

The program generated in the way described above is shown in fig. 6. The additional assumption made by the system is that to the first point of each separate polyline the robot moves with the PTP (Point-To-Point) interpolation and stops there. At that point, according to the specified extended data, glue starts to be fed. The point is given in the absolute measure (key word MOVE) in the robot's base coordinate system. To the following points - given as increments (key word MOVE_REL) - the robot moves continuously (key word VIA) with the linear interpolation and stops at the last point of that polyline. At this point the glue feeding is stopped.

After the simulation is started, it can be monitored in debugger-like mode - one can run the program step by step, set/reset the breakpoints and collision detection, display the robot's state and so on. All the necessary information about the simulation (robot's position, effector state, collisions, time etc.) is stored in a disc file and can be analysed after the simulation is finished. The video frames (fig. 8) are also stored and can be displayed as a film using an animation software.

The picture below (fig. 7) presents the workcell modelled using the RME System. The workcell consists of one DFG robot performing the gluing operation and a car body to be glassed. After the body has arrived at the station the robot starts to apply glue along the window frame. This frame is extracted from the body car drawing and defined as a continuous polyline. Using the program generator described above additional data was associated with some vertices of that polyline. During simulation the robot follows the polyline according to the specified extended data.

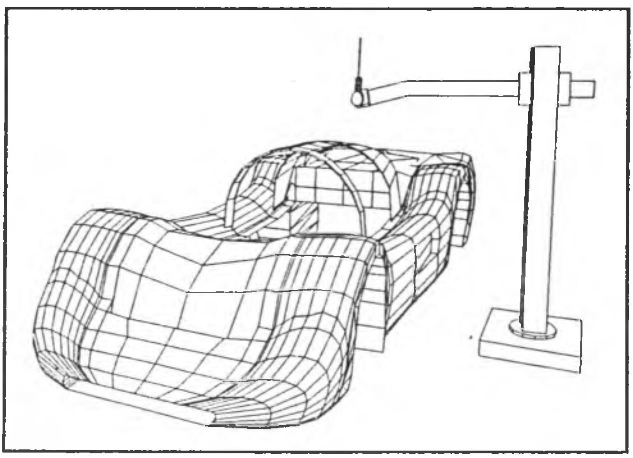

Fig. 7. A view of the robotised workcell modelled within the RME System

angles defined by the user. If the orientation at some points is not defined by the user, the system approximates it by linear interpolation according to the nearest points, where the orientation was defined earlier. In the next step the trajectory has to be inserted into the workspace of the robot. For the current position and orientation of the trajectory the transition matrix between the local coordinate system of the polyline and the local coordinate system of the robot is calculated by the system. Then, the generator reads all the parameters associated with the given polyline (motion parameters, effector state and orientation) and generate the program in BAPS - the Bosch language for robots (Ger.: Bewegungs- und Ablaufprogrammier Sprache).

The final result of the program generator is a disc file with the program for the robot. Before the program is simulated, it must be compiled into the time-optimal intermediate code - within the RME system it is the IRDATA code.

```
PROGRAM demo

; external programs declaration
EXTERN: glue_on, glue_off

; global interpolation
;;INT=LINEAR

; velocity factor
VFAKTOR=0.1

; move absolutely to the first point with PTP interpolation
MOVE PTP TO (1418.27,405.58,668.77,90.00,0.00,0.00)

; call external program for feeding glue
glue_on

; move relatively with linear interpolation to the following points and do not
; stop there
MOVE_REL VIA    (25.33,-0.29,205.81,27.25,0.00,0.00),
                (71.48,0.00,133.30,19.88,0.00,0.00),
                (110.84,0.00,112.98,20.80,0.00,0.00),
                (143.41,0.00,87.32,22.07,0.00,0.00),
                (169.21,0.00,56.31,0.00,0.00,0.00),
                (6.11,-540.25,0.39,0.00,0.00,0.00),
                (-3.81,-269.93,-16.57,0.00,0.00,0.00),
                (-164.25,0.00,-46.40,-18.05,0.00,0.00),
                (-139.21,0.00,-71.95,-16.57,0.00,0.00),
                (-107.59,0.00,-93.10,-15.05,0.00,0.00),
                (-69.38,0.00,-109.84,-13.74,0.00,0.00),
                (-24.59,0.00,-250.15,-26.58,0.00,0.00)

; call external program for stop feeding glue
glue_off

; stop the program
HALT
END
```

Fig. 6. The program generated for a robot gluing the car window

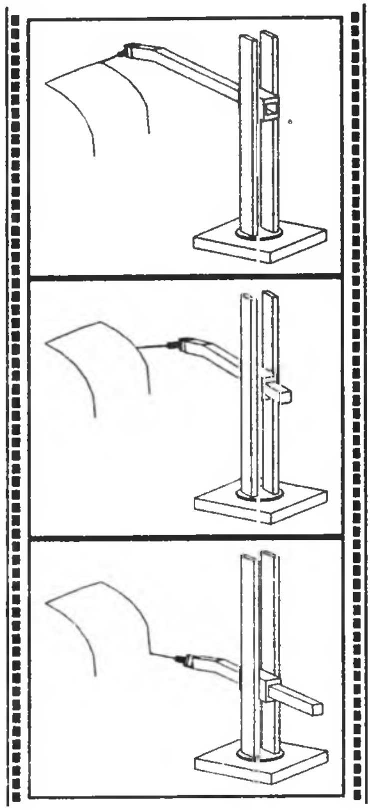

Fig. 8 Film frames captured during the simulation of the robot performing the gluing operation

REFERENCES:

[1] Koch T., Dybala B., Kowalczewski P., "Robot Workcell Simulation System for Personal Computers", International Conference on Computer Integrated Manufacturing CIM'92, Zakopane, Poland, March 1992.

[2] Koch T., Kowalczewski P., Kocełuch A., Dybała B., Szypicyn A., "System symulacji gniazda zrobotyzowanego" IV Krajowa Konferencja Robotyki, Wrocław, Poland, September 1993.

[3] Pritschow G., Bauder M., Angebauer R., "RDL - ein Werkzeug zur Robotermodellierung", Robotersysteme 7, 1991.

[4] Dilmann R., Schneider S., "Ein CAD-unterschtützter Trajektorien Entwurfseditor", Robotersysteme 4, 1988.

Revised by: Gabriel Kost