ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1992
Seria: MECHANIKA z. 108 Nr kol. 1161

International Conference on
COMPUTER INTEGRATED MANUFACTURING
Internationale Konferenz über
RECHNERINTEGRIERTE FERTIGUNGSSYSTEME
Zakopane, March 24-27 1992

Bogdan DYBAŁA, Tomasz KOCH, Paweł KOWALCZEWSKI

Institute of Machine Technology and Automation
Technical University of Wroclaw

ROBOT WORKCELL SIMULATION SYSTEM FOR PERSONAL COMPUTERS

Summary. The paper presents ideas and concepts of a robot workcell simulation system. A method of describing a robot's kinematics in the RDL language is shown. Using RDL allows automatic generation of programs performing forward transformation. Methods of modelling robot and workcell geometry are also presented. The system uses the AutoCAD AME package to graphically animate the workcell.

1. Introduction

Robot machining and assembly workcells make many production processes economical and flexible. In comparison with a fixed concept of automation where a solution to nearly every production task must be considered when designing a complete production line, robot workcells are more flexible since their adaptation to a special task may take place at any time. The workcells have this property thanks to free programmability of their control systems and their mechanical advantages such as good movability, ability to change work tools etc.

The increasing degree of automation and constant tendency towards small batch production cause the increase of cost of programming in the total cost of production. This fact may question economical effectiveness of time-consuming programming methods such as teach-in programming of robots or programming of PLCs in an assembly language. These methods have two important disadvantages:
- expensive production lines must be stopped when the machines need to be re-programmed,
- complicated programming requires long-term testing and debugging of programs and extensive staff training.

New concepts of programming of robot workcells include [1]:
- off-line programming and testing of all units of the workcell,
- self-documenting programming languages,
- general but common programming of all control systems: RC (robot

control), NC (numerical control of machine tools) and PLC (programmable logic controllers),
- ability to use data from CAD systems,
- effective integration of geometrical and technological data at the workpiece.

Off-line programming is especially effective if it may be supplemented by off-line testing of the created programs away from the machines. To achieve this, simulation systems are developed. The systems should enable testing of programs for all workcell units with graphical animation of their movements, checking and optimizing control programs, collision detection etc. They may also be used to design and optimize the workcells. Simulation systems are being developed in academic centres [1-6] and large manufacturing companies [7-12].

2. Simulation system

A draft concept of the robot production workcell simulation system being developed in the Institute of Machine Technology and Automation is shown in figure 1. The software package consists of three parts: modelling of robots, modelling of workcells and actual graphical animation.

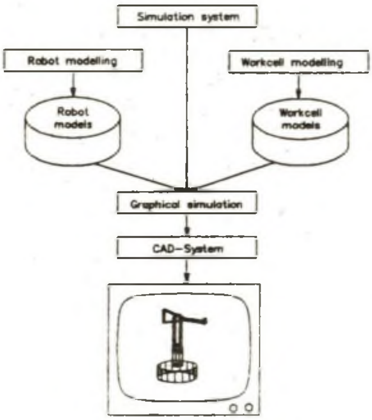

Fig.1. The simulation system

Modelling of a robot's kinematics uses data obtained from compilation of an RDL description (see the next chapter). Modelling of a robot's geometry uses a CAD system - AutoCAD AME.

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 rest of the workcell may be modelled using tools available in the CAD system. A complete workcell model may be stored in another database.

Having created a model of a workcell the user may simulate the workcell graphically. The system will allow simulation of robots controlled manually in both machine and world coordinate systems and of robots controlled automatically by a program - the latter when the system is integrated with a system for off-line robot programming. In order to create an interface between an off-line programming system and a simulation system a problem of the variety of programming languages must be solved. The project being presented will use a solution suggested by many authors of simulation systems [13] - the IRData code. IRData (Industrial Robot Data) [14] is a standardized code independent of a particular robot or its control system and is used by many robot manufacturers as an intermediate code transferred to actual robot controllers [15].

3. Robot kinematics description language

A description of a robot's kinematics in a form appropriate for a computer processing enables automatic generation of a forward and inverse transformations for the robot. Additional data about the robot's geometry enables graphical animation of its movements.

A description of a robot kinematics may be based on Denavit-Hartenberg parameters [16]. Such a description however has a drawback: the parameters are sometimes difficult to imagine and in order to find them the user may need to make some calculations. The authors of the simulation system being presented here have decided to use a clear kinematics description which does not need calculations and the user may find it with a single sketch of a robot.

A language for a robot kinematics description was proposed in the Institute of Control Technology of Machine Tools and Manufacturing Systems of the Technical University of Stuttgart, Germany. The language is called RDL (Robot Description Language) [17]. The presented simulation system utilizes the proposed language and therefore the authors use the original name of RDL.

The figure 2 shows an example robot whose kinematics may be described in RDL as follows:

```
ROBOT TEST
     LINK L1
          POSITION Z=100
     JOINT J1
          ROTATION Z+
          RANGE FROM -160 TO 160
     LINK L2
          POSITION Z=1580
     JOINT J2
          TRANSLATION Z+
          RANGE FROM -200 TO 200
     LINK L3
          POSITION X=1130
     JOINT J3
          TRANSLATION X+
          RANGE FROM -225 TO 225
```

```
        LINK L4
             ORIENTATION Y=10
             POSITION X=450
        JOINT J4
             ROTATION X+
             RANGE FROM -250 TO 250
        LINK L5
        JOINT J5
             ROTATION Y-
             RANGE FROM -210 TO 30
        LINK L6
        JOINT J6
             ROTATION Z+
             RANGE FROM -270 TO 270
        LINK L7
             POSITION Z=425
   END
```

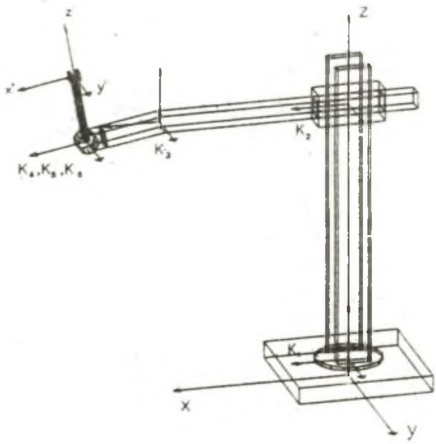

Fig.2. A robot kinematics described in RDL

4. Forward transformation

Basing on a robot's RDL description a forward transformation for the robot may be unequivocally calculated. A forward transformation may be used to compute a position and an orientation of a robot's end-effector as a function of the robot's joint coordinates, i.e. angles and linear displacements of its revolute and prismatic joints. A position of the end-effector is described with a translation vector $t = [x\ y\ z]^T$. Its orientation is described with Euler angles Y, X, Z.

A robot's complete forward transformation T is a product of simple transformation matrices of all links (L_1, L_2, ..., L_n) and joints (J_1, J_2, ..., J_{n-1}):

$$T = L_1 J_1 L_2 J_2 \ldots J_{n-1} L_n$$

A position and an orientation of the i-th link may be found using the same simple transformation matrices of all preceding links and joints:

$$T_i = L_1 J_1 L_2 J_2 \ldots L_{i-1} J_{i-1}$$

Such method of describing a position of each link conforms to requirements of the AutoCAD AME package and simplifies visualising simulated robots.

5. Compiling a robot's RDL description

The simulation system includes an RDL compiler which basing on RDL data generates a C-language code of a program performing a forward transformation.

During the first step of the compilation the compiler reads and analyses the RDL file and creates four files with the following data:

a) MTX - contains matrices of transformations between successive joint coordinate systems,
b) PAR - contains parameters defined in the RDL file, used for initialisation of the simulation system,
c) JNT - contains joint definitions, used for initialisation of the simulation system,
d) ERR - contains error messages if compilation was unsuccessful.

In the second step of the compilation a C-language code is generated. The code contains a time-optimized function computing a forward transformation (a position and an orientation of a robot's end-effector based on all joint coordinates) and a function computing a single transformation for all robot's links (positions and orientations of each joint's coordinate system in respect to a robot's base).

6. Geometrical robot modelling

A geometric model of a robot must be defined in terms of links only. It means that objects modelling the shape of the robot which are responsible for a joint (eg. a motor) must be regarded as a part of either of the neighbouring links.

It must be emphasised here that a geometric model is always a compromise: the more detailed a model the more complicated calculations are required to redraw it, but the less detailed the model the less adequate for problems like collision detection.

Modelling a single link is done with AutoCAD AME (Advanced Modelling Extension). A local coordinate system for a session of designing a link is determined by a coordinate system assigned to the joint preceding the link.

The process of modelling a link is a process of designing a solid body. The body of a link is created as a sum of solid primitives available in AME such as cuboid, cylinder, cone, wedge. The primitives may be combined using boolean operations: sum, union, difference.

7. Modelling a workcell and visualising a robot's movements

The simulation system in its current state may model a workcell with several robots (figure 3), each robot may be visualised in any position and configuration of its machine coordinates. The latter possibility is a direct basis for simulating a robot's movements when the robot is controlled manually in machine coordinates.

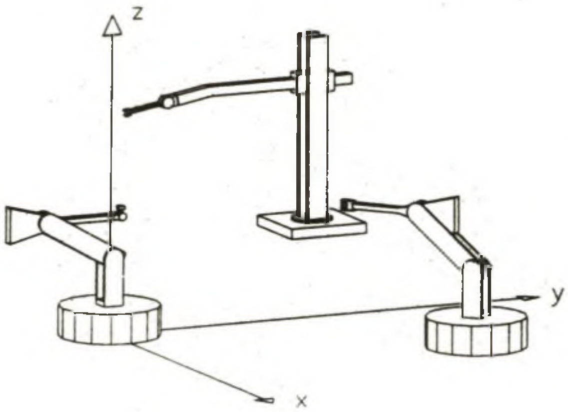

Fig.3. Workcell with robots

The table 1 shows the insertion points of the robots in the workcell from figure 3, the table 2 shows machine coordinates of the robots.

Table 1

	X	Y	Z	rotX	rotY	rotZ
ROB1	0	0	0	0	0	0
ROB2	2000	3000	0	0	0	60
TEST	-4000	4000	0	0	0	0

Table 2

	ROB1	ROB2	TEST
m1	0	70	-90
m2	60	-50	-200
m3	-60	50	0
m4	0	90	0
m5	0	0	-90
m6	0	0	0

8. Planned development of the simulation system

Currently the simulation system can model a single robot: its kinematics and geometry, and a workcell with several robots. Future plans of development include:

1 - problems of robot kinematics
  - algorithms for automatic generation of inverse transformation with iterative methods
  - algorithms for verifying programs calculating forward and inverse transformations
  - modelling of robot-environment interactions (eg. picking a workpiece)

2 - simulating robot control systems
  - PTP movements control
  - linear approximation movements control

3 - interfacing the simulation system in the IRData standard with professional systems for off-line robot programming (the Institute owns a system for off-line programming of BOSCH robots)

4 - modelling of robot dynamics

5 - collision detection in a workcell.

REFERENCES

[1] Pritschow G., Frager O., Schumacher H., Wieland E.: Programmierung von roboterbestückten Produktionsanlagen, Robotersysteme 5 (1989), Springer-Verlag

[2] Schumacher H.: Einheitliche Programmierung von Automatisierungskomponenten roboterbestückten Bearbeitungs- und Montagezellen, ISW Uniwersytet w Stuttgarcie

[3] Leu M.C.: Robot Motion Simulation and Planning Based on Solid Modelling, Annals of the CIRP Vol. 37/1/1988

[4] Dai F., Dillmann R.: Computergraphik für die Roboterprogrammierung, Informatik Forschung und Entwicklung 3 (1988), Springer-Verlag

[5] Weck M., Niehaus Th., Osterwinter M.: An Interactive Model Based Robot Programming and Simulation Workstation, A. Storr, J.F. McWoters (Editors), Elsevier Science Publishers B.V. (North-Holland), IFIP, 1987

[6] Wloka, D. W.: Robotersimulation, Springer-Verlag, Berlin 1991

[7] Weck M.: Werkzeugmaschinen Band 3: Automatisierung und Steuerungstechnik, VDI-Verlag, Düsseldorf 1989

[8] Spur G.: Stand der Programmiertechnik für Industrierobotern, Fertigungstechnisches Kolloquium in Stuttgart, Springer-Verlag 1988

[9] Craig J.J.: Introduction to Robotics, Addison-Wesley Publishing Company 1989

[10] Simulation vermeidet Fehler, Roboter, Oktober 1989, Verlag Moderne Industrie

[11] Off-line auf dem Punkt, Roboter, September 1989, Verlag Moderne Industrie

[12] Durchgängige Simulation, Roboter, November 1989, Verlag Moderne Industrie

[13] Dillmann, R.; Huck, M.: Informationsverarbeitung in der Robotik, Springer-Verlag, Berlin Heidelberg 1991

[14] VDI 2863: IRDATA - Programmierung numerisch gesteuerter Handhabungseinrichtungen - Allgemeiner Aufbau, Satztypen und Datenübertragung, Düsseldorf: VDI-Verlag 1987
[15] Robotersteuerung Bosch rho 2. Handbuch, Robert Bosch GmbH, Erbach 1986
[16] Denavit, J.; Hartenberg, R. S.: A kinematic notation for lower pair mechanisms based on matrices. ASME J. Appl. Mech. 22 (1955) 215 - 221
[17] Bauder, M.; Schumacher, H: Allgemeine Lösung der Koordinatentransformation für Industrieroboter. HGF-Kurzberichte 87/6, Industrieanzeiger 109(1987) 27-28

SIMULATIONSSYSTEM FÜR DIE ROBOTERZELLE

Zusammenfassung

Dieser Beitrag beschreibt das im Institut für Fertigungstechnik und Automatisierung der Technischen Universität Wroclaw für das Personal Computer entwickelte Simulationssystem für Roboterzelle. Es wurde die Sprache für die Beschreibung der Roboterkinematik und ihre Übersetzung, die automatische Vorwärtstransformation ermöglicht, vorgestellt. Die Generierung von geometrischen 3D Modellen für Robotern und Roboterzellen sowie grafische Darstellung von Bewegungsabläufen erfolgen mit Hilfe AutoCAD AME Systemes. Zum Schluß sind geplante weitere Module des Simulationssystemes vorgestellt.

SYSTEM SYMULACJI ZROBOTYZOWANEGO GNIAZDA WYTWÓRCZEGO

Streszczenie

Praca przedstawia idee i koncepcje opracowywanego systemu symulacji zrobotyzowanego gniazda produkcyjnego. Zaprezentowano sposób opisu kinematyki robota w języku RDL. Zastosowanie języka RDL umożliwia automatyczne generowanie modułów programowych obliczających transformacje proste. Opisano modelowanie geometrii robota, modelowanie gniazda produkcyjnego oraz wizualizację ruchu robota. Do graficznej animacji gniazda system wykorzystuje pakiet AutoCAD AME.

Wpłynęło do redakcji w styczniu 1992 r. Recenzent: Jan Kaźmierczak