Assoc. Prof. Ing. Jozef N. Marcinčin, PhD. - Ing. Pavol Petruška
Technical University of Košice, Prešov, Slovakia

Facilities of Automatized Manufacturing Systems Simulation by Special Simulation Software

Abstract: *At necessity a computer adhesion for design, animation, analysis and simulation of automated manufacturing systems not misgiving. Together existing general spectrum CAD pruduct only for design and 3D animation, example: I-DEAS, UNIGRAPHICS, Pro/ENGINEER, CATIA, etc. It are a big complex systems will run on workstation and trough are highly power bat capital expediture are high, also above dissalled kinematic and dynamic analysis, calibration and monitoring purposes. On the other hand to be special softwares, which make use of this purpose too. It are the unigue PC based analytical and simulation programming systems, example: ROANS, ROBCAD, KUSIM, etc. The paper describes how the ROANS software, that oneself often utilized at building of the automated systems and simulating the proces.*

1. INTRODUCTION

The ROANS is analytical and simulation system, which has next characteristics:

- is a PC based modular software package, dedicated to the analysis of automated manufacturing systems, preferential with industrial robots,
- offers tridimensional simulation, kinematic and dynamic analysis of manufacturing systems with industrial robots and engineering support of new automated devices design,
- the controller of any mechanical system the user has graphically created - peermits to rapidly evaluate several different layout solution, reducing errors and allowing a wider range of possible choices,
- can help engineers dimension motors and gear reducers, optimize the dynamic behavior of a mechanical system [1].

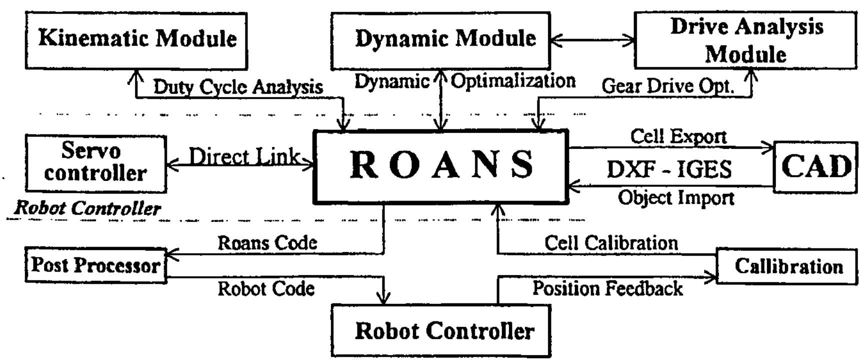

Fig. 1 ROANS Configuration

The ROANS Configuration is presented on Fig. 1.
ROANS operates in five conceptually different areas:
1) *Building of the automated cell.* The user can create any object in a 3D editor or import if from CAD (DXF, IGES), then compose the cell by listing the object in it, and, finally, verify if the devices reaches everywhere in the cell without interfering with any object.
2) *Programming and simulating the proces.* ROANS is a high level language, working in an interactive way with the graphic representation of the cell. Kinematic routines calculates the cycle time and verify for interferences and collissions. The user can verify be correctness of his multi-tasking programming logic by simulating inputs and PLC communications.
3) *Analysis the data.* All the information concerning position, velocity and acceleration is presented by the kinematic module. Inertias, moments, internal forces, RMS power, dutycycle, and other motor parameters are calculated in the dynamic and drive analysis module.
4) *Communicating with the automated workcells.* Several post-procesor modules are available with the goal of translanting ROANS language into the robot code and to translate the robot language into ROANS code. ROANS has routines to calibrate both the cell and the robot itself.
5) *Direct programming.* A system has been created in which ROANS operates as programming unit and trajectory generator for a particular axis controller board. The user can graphicaly create a complety new device and ROANS will act as a controller. With ROANS directly accessing the controller memory, the positive aspect of on-line and off-line programming are summer into direct programming, a world in which reality and virtuali come together. The view to screen of the ROANS is on Fig. 2.

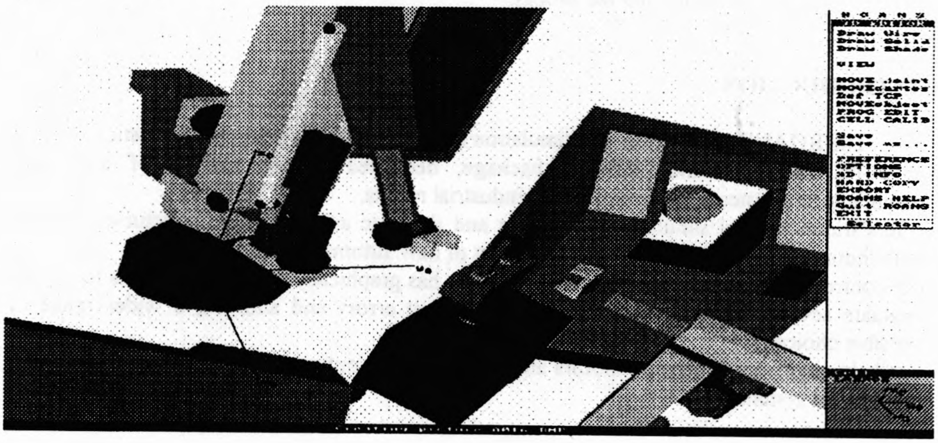

Fig. 2 View to screen of the ROANS

2. DATA STRUCTURES IN ROANS

Data in ROANS are represented by two base structural types, the workcell and objekt type. The workcell consists of object, whereas the object is an elementary entity, futher not structured. These data structures are beging created and updated within the ROANS OBJECT and ROANS CELLS [2].
ROANS CELLS option still allows the user to write the programs to control the workcell and animate it in the 3D graphics.

ROANS OBJECTS - There are five object types in ROANS, every characterized by a specific data set. Static object is an object with fixed position in its reference space, that is to say, the static object can not be moved by robots or by any other manipulating systems. A table, fixed to the ground, is a typical example. Dynamic object is an object, its position within the reference space can be changed by robot. The dynamic object is assigned the grip location, in witch the object is grasped during the object manipulation. A typical examples are parts for assembly, parts stacked on pallets, etc.

Tool is a single link end-efector mounted on the mechanical interface of a robot at the TAP (tool attachment point) or on machine tools, etc. The TCP (tool center point) is assigned to each tool, specifying the technological point of the tool. The tools have also assigned a grip location so that the tool may be gripped and mounted onto the mechanical interface at the TAP. Gripper is the end-efect of a robot, represented by a multi link mechanical system, attached to the mechanical interface of a robot at the TAP. The TCP is assigned to each gripper, specifing the technological point of the gripper.

MECHANICAL SYSTEM is represented by any multi link system with links connected by joints. A typical example is robot or machine tool. The motion of mechanical systems can be programmed in high level ROANS language, or can be animated manually. ROANS supports modeling of the following types of special mechanical systems:

- simple open-loop mechanical systems S,
- tree structured mechanical systems T,
- closed-loop mechanical systems containing planar kinematic loops C [2].

The mechanical system is structurally described as the kinematic chain with the base link fixed in the workcell space. Before the mechanical system will be referenced as a workcell object, must be created first and stored into the ROANS object database.

3D GRAPHIC - The 3D graphic editor of ROANS provides the user with elementary editing tools for creating the 3D models. However, user may use also his favorite CAD system to can be the 3D models imported into ROANS (DXF and IGES protocols) [3]. The 3D models of object are represented by their surfaces, composed of the planar faces. The 3D graphic model of the mechanical system is based on the 3D graphic representation of its links at one of the three drawing commands: Draw Wire, Draw Solid, Draw Shade.

The topology description of mechanical system is based on the graph theory. For every kinematic chain there is a class of indirected graphs, defining the structure of mechanical system and the link incidence between the links of the system. The structural description represents the primary input that enables ROANS to recognize the structural type of the mechanical system and to control the computational precedence within the kinematic and dynamic simulation, graphics modeling, etc.

The topology of a mechanical system is defined by the path matrix of the graph, assigned to the mechanical system. The lines of the path matrix specify the paths leading from the base link to the pendant vertices of the graph. Pendant vertices are the end vertices of the graph. The topology description is common for any multi body mechanical system regardless of its structural type, Fig. 3.

The structure, virtual or no, is characterized by number of graph paths (branches of the tree) and by the length of the longest path of the graph - Maximum graph path lenght.

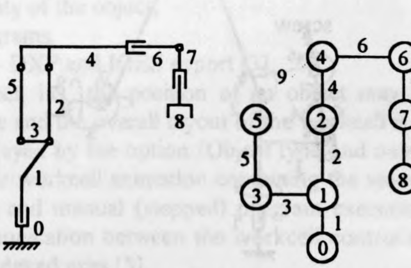

Fig. 3 Example of the graph assignment to closed loop mechanical system of the robot

Besides the rules formulated in preceding chapter there is one more that regards the arrangement of lines of the path matrix of the tree graph: every line of the path matrix represents the graph path, its pendant index is higher than that one of the path from the preceding line. The path matrix may contain max. 6 path (lines), each with two links at least. Note that the maximum number of D.O.F. for any mechanical system is limited to 6 [2].
The standard library of ROANS supports most of the existing structural types of robots, that supported by the inverse kinematic transformation and may be programmed to follow prescribed trajectories, for example, the linear or circular trajectories. Robots missing the inverse transformation may be programed just in the PTP control mode with the target position expressed in terms of joint coordinates. S and T structured systems can be created and stored into ROANS data base by the user. New closed loop systems will be instaled into ROANS library on reguest by authors of software, else the user will not miss the inverse kinematic transformation.
ROANS support modeling of kinematic pairs off fifth class, i. e. with one degree of freedom - D.O.F.. Every type of the kinematic pairs is being referenced by its identifier: R-rotational, T-translational, H-helical, C-cylindrical and S-spherical kinematic pairs. A new kinematic notation has been introduced in ROANS to support the kinematic modeling off all types of kinematic chains, while still supporting the Denavit-Hartenberg orthogonal notation (submatrix 3x3 of the Denavit-Hartenberg homogenous transformation matrix) generaly accepted in robotics. There are closed loop system and tree like systems, for which the notation is not valid any more. Therefore called j notation, has been introduced in ROANS to support all types of kinematic chains (is defined by three subsequent rotations of a coordinate frame around the z, x and z axis by angels, and respectively).
ROANS comes with a new compututational scheme based on the Equivalent Effect Principle which is a new concept in the teory of dynamics of the mechanical systems containing the closed kinematic loops [4].

The closed loop mechanical systems are internaly cut in the kinematic loops, so that the virtual trees, also referred to as the equivalent trees, are created. The joint, in with the kinematic loop is cut, is called cut joint. As the creating of closed loop system with new structures is not allowed to the user, then the term 'cut joint' will not be closely described hereon. There are two categories of joints of the mechanical systems introduced in ROANS. The first category is represented by the active and passive joints. The active joints are actuated

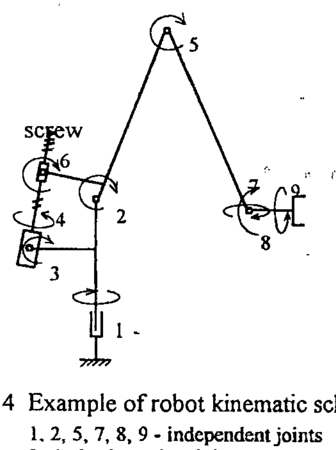

Fig. 4 Example of robot kinematic scheme

1, 2, 5, 7, 8, 9 - independent joints
3, 4, 6 - dependent joints
1, 4, 5, 7, 8, 9 - active joints
2, 3, 6 - passive joints

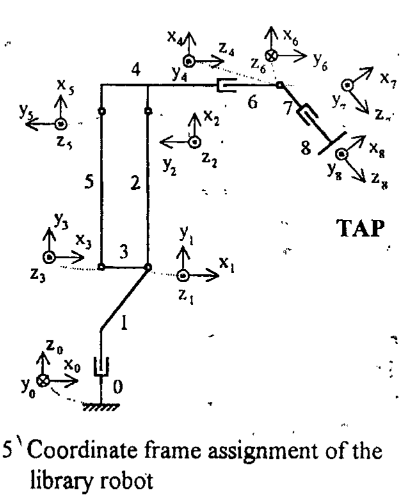

Fig. 5 Coordinate frame assignment of the library robot

by drives, the passive joints are driven. The second category is represented by the so-called independent and dependent joints - Fig. 4.
Important is enable the user to define his own zero position and motion signature of the robot joints - Joint reset, Joint (Actual) Coords and his own position and orientation of base coordinate the frame of the mechanical system (x, y, z, Rx, Ry, Rz) - Base reset, Cartes Coords. The example is as shown on Fig. 5.
KINEMATIC INPUTS - allows the user to define the metrics, TCP notation, velocity and acceleration limits of joints of the mechanical system. The metric defines relative position of incident links in the mechanical system and is defined by the offset vectors r_{ij}. Every ROANS object is assigned a triple of calibration points, which are represented by the triple of position vectors. For the mechanical systems the vector defining the position of the triple of points are expressed in terms of coordinates of the link, in space of which the calibrations points are defined. In addition to this, the index of the link, in space of which the calibration point are defined, is to be specified within this option too. TCP notation defining the orientation of the TCP witch respect to TAP, or witch respect to a reference coordinate system. ROANS supported the Bryant and RPY notation. ROANS supports the inverse dynamics and drive system analysis of the mechanical system, providing thus possibility for optimization of control parameters with respect to the control stability, power capacity of actuators provided the DYNAMIC INPUTS must be described - masses, mass center and central tensor of inertia.
DRIVE INPUTS - allows the user to select the motors and speed reducers from the ROANS database and assign the drives to the active joints. Within the drive system analysis the input joint forces and/or torgues needed to drive the system along the programmed trajectory are computed. In addition to this the motor torgues are computed too, taking into account the Coulomb and viscous forces in the joint transmission. For the gear reducer and motors the effective torgues, related to the programmed duty cycle, are computed too.
CONTROL - Within this option the type of control of the active joints is specifield, that is to say, every active joints is classifield to be either the servoed or indexed axis.For the servoed joints, the encoder pulses instaed of degres or millimeters can be enabled as the joints position units. It means, the position of robot joints is programed and monitored on the screen in pulses.
WORKCELL - is the entity of highest structural level and consists of object, their position is either fixed or changing during the process simulation. After the loading of existing workcell, or after the specification of the name of a new workcell, the workcell data and the workcell layout can be (re)defined. After gets the program control to a multilevel self documenting menu system that provides the user folloving options:

- view of the workcell in various graphic modes,
- moving and manual control of the workcell objects,
- checking the object collision and accessibility of the object,
- graphic interactive editing of ROANS programs,
- position calibration of the workcell object - DXF and IGES export [3].

Object in ROANS may have their reference object, i.e., the position of an object may be defined in the space of another object. The structure and the overall layout of the workcell is to be defined within the workcell structure table displayed by the option (Object type and name, branch and object index, master, type of slave). The workcell animation containing the second pass ROANS compiler and options the automatic and manual (stepped) program execution. The purpose of the cell I/O is to define the communication between the workcell controllers (master program) and betwen the controllers and indexed axes [5].
KINEMATICS OUTPUTS - by usign the joints positions, velocities and accelerations of selected mechanical system can be visualized on the display (q, $\dot{q}$, $\ddot{q}$).

DYNAMIC and DRIVE OUTPUTS - by using the joint forces, torgues. input generalized forces and motors torges, current and voltages, need to drive the mechanical systems along the programmed trajectories may be plotted on the display or a peripheral device.

3. CONCLUSION

ROANS application covers the 3D graphic animation, programming and design of robots, planning and controlling of workcells, commercial presentation and education.
Complex task of the 3D graphic simulation and programing of robots and workcells can be carried out by ROANS on PCs with the performance equivalent to the workstation based systems.

4. ACKNOWLEDGEMENTS

This work was supported by Slovak Ministre of Education Research Project, contract 1/4367/97 "Research and Development of Technical Systems Based on Biomechanisms for Humanization and Handicaps Elimination of Physically Disabled People". The software ROANS was render for authors this paper by author of ROANS - Ing. Martin Doliak (Presov, Slovakia).

LITERATURE

[1] Janek, B. - Fecko, T. - Doliak, M.: ROANS - Advanced PC Based Simulation and Programming System for Robots and Automated Workcell.In: Proceedings of the International Conference ROBTEP - Robotics in Theory and Practice, Presov, 1993, pp. 195 - 198.

[2] ROANS v. 2.0. Reference manual, JHF s.r.o. Presov, 1993.

[3] Pavlikovic, M.: Preprocesor of Graphic Protocol IGES for Simulation in System ROANS. Diploma Work. Technical University of Kosice, Presov, 1994.

[4] Janek, B.: Equivalent Effect Principle - a new approach in Closed Loop Manipulators Dynamic. In: Proceeding of the 5th I.C.A.R. Conference, Pisa, 1991.

[5] Doliak, M.: Off-line Programing of Robots. Diploma Work. Technical University of Kosice, Presov, 1992.

[6] Petruska, P. - Marcincin, J. N. - Doliak, M.: Simulation of robots and robot cells by software ROANS. In: Proceedings 6th International Workshop Robotics in Alpe - Adria - Danube Region RAAD97, Cassino, 1997, pp. 296-300.