

# A Reverse Engineering Technique for Creating Virtual Robots

Simon Brezovnik\* - Miran Brezočnik - Simon Klančnik - Ivo Pahole - Karl Gotlih  
University of Maribor, Faculty of Mechanical Engineering, Slovenia

*A novel system for out-of-date robotic systems that are not interesting for use any more is introduced. We give a solution for revitalization of these robotic systems with upgrading the programming software from on-line programming to indirect virtual programming and virtual modelling of a production cell equipped with robots. Direct and inverse kinematic models of the robot ACMA XR 701 in the virtual space were developed with reverse engineering technologies.*

© 2009 Journal of Mechanical Engineering. All rights reserved.

**Keywords: robotics, robot modelling, reverse engineering, robot cell**

## 0 INTRODUCTION

The introduction of robots into modern industry is strongly connected with production automation where robots are used for material and parts manipulation and direct manipulation of tools in a technological process. Implementation of robots decreases production costs, increases productivity, assures the quality of production, and can replace employees at hard and dangerous operations. Robots in most cases are used for serving production machines, welding, painting, assembling, packing, for manipulation of parts directly in the production, and also for control tasks like grinding and automatic optical control.

The main reason for automation and robotisation is cost reduction, discharge of workers and the assurance of production capacity and quality. The automation and robotisation decrease production time and production costs, and increase production capacity. Unfortunately, the decrease in production costs cannot be calculated and justified easily.

If we look back to the beginning of automation and robotisation, we see that it was stimulated by the requirement for discharging workers of dangerous and monotonous tasks, and the fact that some tasks could not be done efficiently enough.

Companies look at the automation and robotisation from the viewpoint of savings and expenses, and also as an opportunity to remain competitive in their industrial field [1] to [4].

The variety of applications and innovations in the field of robotics dictate the change of older robots with new ones not only because of mechanical properties, but in most

cases for programming reasons. New systems are more efficient and user-friendly [6] to [10]. Therefore, older robots are a problem in companies; they are mechanically suitable, but are taken out of the production for programming reasons.

In this paper we suggest an approach to upgrade the programming software of a robot ACMA XR 701 from Renault to the level of a virtual model with the use of reverse engineering (digitalization). With the virtual model we get the opportunity of indirect programming and modelling of the robotized production cell in a CAD modelling system. For the virtual robot model we developed direct and inverse general kinematic models with six degrees of freedom.

## 1 DATA PREPARATION FOR DIGITALIZATION

For best fitting of the robot model with the real robot we used a special equipment for digitalization which is installed in the Laboratory for industrial design at the Faculty of Mechanical Engineering in Maribor. This portable equipment performs high resolution and excellent accuracy.

The digitalization process was used because it satisfied the requirements:

- non-contact digitalization,
- no constraints on shapes, weight and materials,
- no requirements on object set-up,
- high resolution,
- high accuracy,
- easy transportability.

For the acquisition of coordinate points, (Fig. 1), we used an optical non-

\*Corr. Author's Address: University of Maribor, Faculty of Mechanical Engineering, Smetanova 17, 2000 Maribor, Slovenia, simon.brezovnik@uni-mb.si

contact measuring system TRITOP. This measuring system is designed for the acquisition of coordinate points at 3D objects. For high quality surface recognition at least five reference points had to be stuck on each surface.



Fig. 1. Coded reference points

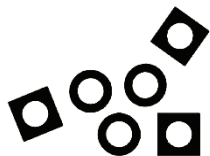


Fig. 2. Non-coded reference points

The technology used works with a photogrammetric digital camera Fuji FinePix S3 that performs a resolution of 12 Mega pixels (4256 x 2848).

After acquiring the reference points that are used to determine the centre of rotation between the axes (Fig. 3), we used a measuring system for digitizing ATOS II 400 for surface detection. This measuring system is a product of the GOM company.

The system works with two cameras and a projector. The pictures obtained were processed with triangulation. In the next step we got a 3D model of the scanned object. The accuracy of the model depended on the focal length of the lens in the objective. The accuracy also depends on the quality of the lenses, the camera resolution, and the wave length of the light, which are all out of user's control.

## 2 THE DIGITALIZATION PROCEDURE

The digitalization procedure was split into several tasks:

- object preparation,
- equipment preparation,
- digitalization, and
- data processing.

The robot surface was cleaned of dust and other particles before the treatment. After that we stuck the coding and non-coding reference points at the robot surfaces. The non-coded points (Fig. 2) were used in the digitalization process, the coded reference points (Fig. 1) were used for assembling snaps, which were taken with the photogrammetric camera. The system for snaps recognition distinguishes and assembles the coordinates of the coding points that can be seen in more than two snaps. It was important to combine snaps taken from different angles and to always collect at least three coding points from the previous snaps.

In this way we got the coordinates of all the recorded points, which were at the same time a basis for digitalization of the robot (Fig. 3).

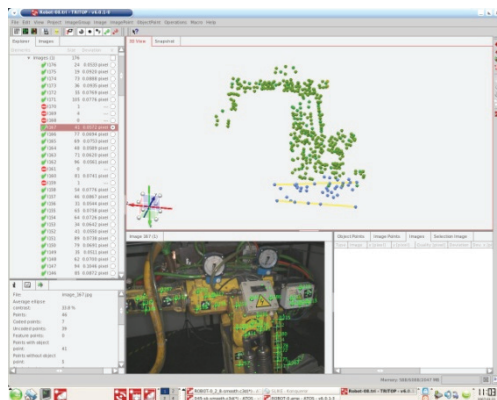


Fig. 3. Reference points

For further digitalization we painted the robot surfaces with a thin layer of white colour to protect the surfaces against light reflection.

We put lenses that are used for acquisition of the measuring volume with dimensions of 135 x 108 x 95 mm with the accuracy of 0.01 mm into the measuring system. The measuring system for digitalization was set on a stiff and stabile portable frame, which enables high quality acquisition of snaps.

Before shooting, the system was calibrated with a special calibration tablet. The original tablets were too small for the dimensions of the robot, so we had to make out a new calibration tablet with proper dimensions for the robot. (The workspace of the robot has a diameter of 7 m.) The new tablet was calibrated with respect to original reference calibration tablets.

The points we got from the photogrammetric camera were transferred into the software ATOS. The non-coded points were a reference for the points obtained with the digitalization.

In digitalization we changed the position of the measuring system after each snap, so we were able to collect all snaps, which were automatically assembled together after the accordance of non-coded reference points. The coded points were not taken into consideration. The result of the assembling process is a cluster of points, which represented the surfaces of the robot (Fig. 4).

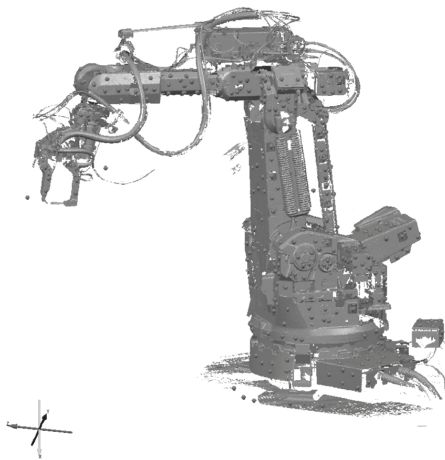


Fig. 4. Cluster of points on robot surface together with reference points

The next operation in the digitalizing procedure was equalization. The equalization of points in the clusters from different snaps was automated but also manual. Manual equalization was used when we did not have enough non-coded reference points. The assembling process was executed with the function of best fitting. This function is normally included in the software that is added to the measuring system, but we can also find it in some special 3D software system for acquisition.

The following step was the data fusion (polygonisation). This step is used for assembling snaps into a mesh. The mesh must be additionally treated. This part of the process is fully automated.

The cluster of points contains noise: points outside of the recorded object (the consequence of the measuring errors) and some unwanted

objects (the table on which the object stands, handles). The removal of the first one is fully automated with some special mathematical algorithms; the second one must be removed manually. Next comes cleaning of the mesh to get a smooth and analogue mesh. The tools for mesh corrections use gap filling and removal of edges [12] to [14].

The aim of the digitalization was to obtain an exact graphical model of the robot, so we imported the meshes in the STL data form into the program Catia V5 R17. Here the points were used as references in robot modelling.

A result of modelling is shown in Fig. 5. Constraints on rotations for all six axes were added to the model and so the dimensions and the shape of the robots workspace could be simulated. As we can see in Fig. 5 the details are not modelled exactly because they do not influence the geometry, which was used for the kinematic robot model. All parts of the robot mechanism, which are important and can be responsible for possible collisions with objects in the environment (collision avoidance), were modelled.

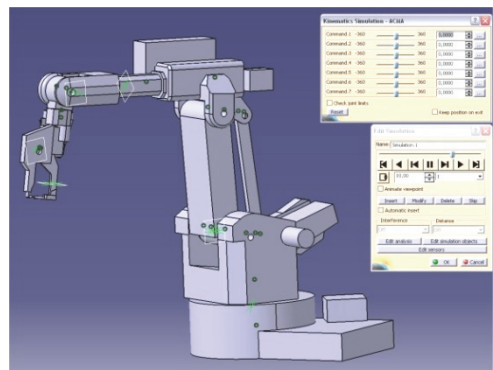


Fig. 5. ACMA robot model in the program Catia V5 R17

With this step the graphical interface was completed. The next step was the preparation of input and output data about rotations of all robot axes. These data are important for the direct and inverse kinematic models.

### 3 DIRECT AND INVERSE KINEMATIC MODELS

The purpose of this paper is to stress the applicability of derivation of the direct and inverse kinematic models by means of reverse

engineering to a wide range of robots. Thus, the direct and inverse kinematic models were initially developed for a general robotic system with 6 degrees of freedom such as a robot Motoman HP6, which is presented in the paper. Later the general model was adapted to the restrictions of the robot ACMA XR701, for which we wish to elaborate a kinematic model by means of reverse engineering.

To describe the position and the orientation of the robots TCP (tool centre point), we had to set a basic coordinate system, which was fixed, and several coordinate systems, which were connected to the moving links on the robot. The rotations or translations in the joints caused relative motion of these coordinate systems, so it was necessary to calculate the transformations between them [23] to [28]. For the description of the rotations and/or the translations we used homogeneous transformations according to Denavit - Hartenberg (DH) [29].

The main idea of the DH notation is the possibility to describe the position and orientation of the TCP with a single 4 x 4 homogeneous transformational matrix. If we have a mechanism with  $n$  degrees of freedom, the complete mechanism is described with the product of  $n$  DH

matrices where each describes the transformation between the  $i$ -th coordinate system with respect to the  $i-1$  system. When all coordinate systems are defined and set, according to the DH notation, the homogeneous transformation matrix is a product of four simple transformation matrices 1 to 2. Eq. (1) holds for rotational joints where  $\theta_i$  is the joint variable.

We put the parameters for the DH notation, for all six joints into the matrix from the Eqs. (1) and (2). The transformation matrices  $\pi_{j=1}^i A_j^{j-1}$  are given in Eqs. (4) to (9).

The homogeneous transformation matrix represents a direct kinematic solution of the robot system.

Following the guidelines of the DH notation we were able to develop an algorithm for describing the kinematic structure via matrices. The setting of the coordinate systems started with a base frame followed by all other systems to the TCP at the top of the robot. The DH parameters obtained for a robotic structure with six degrees of freedom (Motoman HP6) (Fig. 6) are given in Table 1.

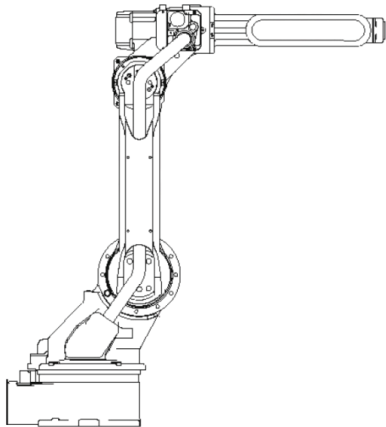


Fig. 6. Motoman HP6 robot

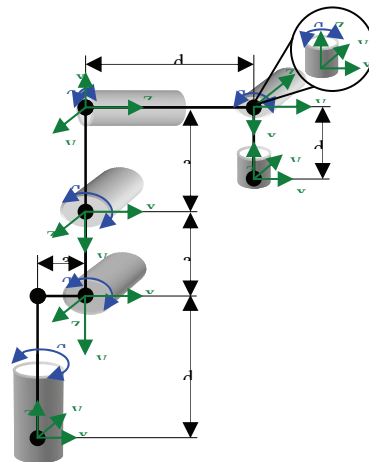


Fig. 7. Setting of coordinate systems on the Motoman HP6 robot with six degrees of freedom

$$A_i^{i-1} = \text{Translation}(z, d) \cdot \text{Rotation}(z, \theta) \cdot \text{Translation}(x, a) \cdot \text{Rotation}(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \quad (1)$$

$$\begin{bmatrix} \cos \theta_i & -\cos \alpha_i \cdot \sin \theta_i & \sin \alpha_i \cdot \sin \theta_i & a_i \cdot \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cdot \cos \theta_i & -\sin \alpha_i \cdot \cos \theta_i & a_i \cdot \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) \quad A_5^4 = \begin{bmatrix} \cos(q_5 - 90^\circ) & 0 & \sin(q_5 - 90^\circ) & 0 \\ \sin(q_5 - 90^\circ) & 0 & -\cos(q_5 - 90^\circ) & a_5 \cdot \sin(q_5) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Table. 1. DH parameters

Robot joint	$\theta_i$	$d_i$ (mm)	$a_i$ (mm)	$\alpha_i$
1	$q_1$	$d_1$	$a_1$	$-90^\circ$
2	$q_2 + 90^\circ$	0	$a_2$	$0^\circ$
3	$q_3$	0	$a_3$	$-90^\circ$
4	$q_4$	$d_4$	0	$90^\circ$
5	$q_5 - 90^\circ$	0	0	$90^\circ$
6	$q_6$	$d_6$	0	$0^\circ$

$$A_6^5 = \begin{bmatrix} \cos(q_6) & -\sin(q_6) & 0 & 0 \\ \sin(q_6) & \cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

The direct kinematic model is represented with matrix  $T_i^0$  [15] to [18]. It is determined with respect to the base frame (3). The final matrix of the direct kinematic model is a product of matrices defined in each joint, which are given in Eqs. 4 to 9. The final matrix of the kinematic model is in Eq. 10.

$$T_i^0 = A_1^0 \cdot A_2^1 \cdot \dots \cdot A_i^{i-1} = \prod_{j=1}^i A_j^{j-1} \quad (3)$$

For  $i=1, 2, \dots, n$ ,  $n$  is the number of degrees of freedom

$$A_1^0 = \begin{bmatrix} \cos(q_1) & 0 & -\sin(q_1) & a_1 \cos(q_1) \\ \sin(q_1) & 0 & \cos(q_1) & a_1 \sin(q_1) \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$A_2^1 = \begin{bmatrix} \cos(q_2 + 90^\circ) & -\sin(q_2 + 90^\circ) & 0 & a_2 \cos(q_2 + 90^\circ) \\ \sin(q_2 + 90^\circ) & \cos(q_2 + 90^\circ) & 0 & a_2 \sin(q_2 + 90^\circ) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$A_3^2 = \begin{bmatrix} \cos(q_3) & 0 & -\sin(q_3) & a_3 \cdot \cos(q_3) \\ \sin(q_3) & 0 & \cos(q_3) & a_3 \cdot \sin(q_3) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$$A_4^3 = \begin{bmatrix} \cos(q_4) & 0 & \sin(q_4) & 0 \\ \sin(q_4) & 0 & -\cos(q_4) & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$T_0^6 = A_1^0 \cdot A_2^1 \cdot A_3^2 \cdot A_4^3 \cdot A_5^4 \cdot A_6^5 = [a_1 \ a_2 \ a_3 \ a_4] \quad (10)$$

The elements of the matrix are  $a_1, a_2, a_3, a_4$  and they are given in Eqs. (11) to (14).

The Eq. (10) is a solution of the direct kinematic problem. It defines position and orientation of the TCP with respect to the base frame, according to the configuration space coordinates (the angles)  $q_1$  to  $q_6$ .

The opposite problem is the calculation of the configuration space coordinates (angles)  $q_1$  to  $q_6$  from the known position and orientation ( $n, s, a$ ) of the TCP. This type of problem is called the inverse kinematic problem.

The inverse kinematic equations contain transcendental functions so that the development of the solution contains the solving of inverse transcendental functions. These functions often do not have a unique solution on the definition area so we have to use a substitution on the basis of the function  $\arctan 2[y/x]$ , which is defined with Eq. (15).

The inverse kinematic model can be derived from the direct kinematic model. The

solution is developed with a recursive algorithm and is shown with Eq. (16).

In most cases we search for configuration space coordinates that are derived from the known position and orientation of the TCP. The elements of the matrix on the left hand side in (16) are known, so we are looking after the variables ( $q_1$  to  $q_6$ ) on the right hand side.

The procedure runs from known configuration space coordinates ( $q_1$  to  $q_6$ ) with solving a system of equations with trigonometric substitutions [19] to [22]. As an example the calculation of the configuration space coordinate  $q_1$  with the Eqs. (17' to (23) is given. The same procedure holds for all other configuration space coordinates.

$$a_1 = \begin{bmatrix} S_1 (C_6 S_4 S_5 - C_4 S_6) + C_1 (C_{23} C_5 C_6 - S_{23} (C_4 C_6 S_5 + S_4 S_6)) \\ C_{23} C_5 C_6 S_1 - C_6 (C_4 S_1 S_{23} + C_1 S_4) S_5 + (C_1 C_4 - S_1 S_{23} S_4) S_6 \\ - C_5 C_6 S_{23} - C_{23} (C_4 C_6 S_5 + S_4 S_6) \\ 0 \end{bmatrix}, \quad (11)$$

$$a_2 = \begin{bmatrix} -C_6 (C_4 S_1 + C_1 S_{23} S_4) - (S_1 S_4 S_5 + C_1 (C_{23} C_5 - C_4 S_{23} S_5)) S_6 \\ C_1 (C_4 C_6 + S_4 S_5 S_6) - S_1 (C_{23} C_5 S_6 + S_{23} (C_6 S_4 - C_4 S_5 S_6)) \\ C_5 S_{23} S_6 + C_{23} (-C_6 S_4 + C_4 S_5 S_6) \\ 0 \end{bmatrix}, \quad (12)$$

$$a_3 = \begin{bmatrix} -C_5 S_1 S_4 + C_1 (C_4 C_5 S_{23} + C_{23} S_5) \\ C_4 C_5 S_1 S_{23} + C_1 C_5 S_4 + C_{23} S_1 S_5 \\ C_{23} C_4 C_5 - S_{23} S_5 \\ 0 \end{bmatrix}, \quad (13)$$

$$a_4 = \begin{bmatrix} C_1 (a_1 - d_4 C_{23} - a_2 S_2 - a_3 S_{23}) \\ S_1 (a_1 - d_4 C_{23} - a_2 S_2 - a_3 S_{23}) \\ d_1 - a_2 C_2 - a_3 C_{23} + d_4 S_{23} \\ 1 \end{bmatrix}, \quad (14)$$

where is:  $S_1 = \sin(q_1)$        $S_{23} = \sin(q_2+q_3)$   
 $C_1 = \cos(q_1)$        $C_{23} = \cos(q_2+q_3)$

$$\arctan 2\left(\frac{y}{x}\right) = \begin{cases} + \arctan \frac{y}{x}; x \geq 0, y \geq 0 \dots I \text{ quadrant} \\ \frac{\pi}{2} - \arctan \frac{y}{x}; x \leq 0, y \geq 0 \dots II \text{ quadrant} \\ -\frac{\pi}{2} - \arctan \frac{y}{x}; x \leq 0, y \leq 0 \dots III \text{ quadrant} \\ - \arctan \frac{y}{x}; x \geq 0, y \leq 0 \dots IV \text{ quadrant} \end{cases} = \arctan 2(x, y), \quad (15)$$

$$T_6^0 = \begin{bmatrix} nx & ox & ax & x \\ ny & oy & ay & y \\ nz & oz & az & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1^0(q_1) \cdot A_2^1(q_2) \cdot A_3^2(q_3) \cdot A_4^3(q_4) \cdot A_5^4(q_5) \cdot A_6^5(q_6), \quad (16)$$

$$(A_1^0)^{-1} T_6^0 = A_2^1 \cdot A_3^2 \cdot A_4^3 \cdot A_5^4 \cdot A_6^5, \quad (17)$$

$$q_1 = MM(y, x), \quad (18)$$

$$q_2 = MM(S_2, C_2), \quad (19)$$

$$q_3 = MM(a_3, d_4) - MM(B, \pm \sqrt{-B^2 + a_3^2 + d_4^2}), \quad (20)$$

$$q_4 = MM(-C_{23}a_y + S_1a_x, S_{23}S_1a_y + C_{23}a_z + S_1C_1a_x), \quad (21)$$

$$q_5 = MM(C_{23}(S_1a_y + C_1a_x) - S_{23}a_z, (S_{23}C_1C_{23}a_x + S_4S_1a_x) + S_1C_{23}S_{23}a_y - S_{23}C_1a_y + C_{23}a_z), \quad (22)$$

$$q_6 = MM\left(\frac{((C_{23}C_1C_4 - (S_{23}C_1C_4S_5 + S_1S_4S_5)o_y + (C_5S_{23}C_4 - S_1C_{23}C_1S_5 + S_4C_1S_5)o_y - (C_5C_{23} + C_4C_{23}S_2)o_z)}{(C_5S_1C_{23} - S_{23}S_5C_1C_4 + S_1S_5S_4)n_x + (C_{23}S_1C_5 - S_5C_{23}S_1S_{23} + S_5S_4C_1)n_y - (C_5C_{23} + C_4C_{23}S_5)n_z}\right), \quad (23)$$

where is:  $S_1 = \sin(q_1)$        $S_{23} = \sin(q_2+q_3)$        $MM = \arctan 2$   
 $C_1 = \cos(q_1)$        $C_{23} = \cos(q_2+q_3)$        $B = d_4S_3 - a_3C_3$

The derived configurations space coordinates ( $q_1$  to  $q_6$ ) are the parameters important for motion synthesis of the virtual robot model. In the case of a direct kinematic model with known configuration space coordinates we can simulate the position and orientation. In the case of inverse kinematic model with a known position and the orientation of the TCP we can simulate the configuration space coordinates and so we get a basic tool for motion synthesis in configuration space coordinates of the robot mechanism in virtual space.

The direct and inverse models that were developed for the general six degrees of freedom robotic mechanism were adopted for the ACMA XR 701 robot. The necessary geometry of the ACMA robot was obtained with the described reverse engineering process. The robot ACMA XR 701 has a parallelogram mechanism (between the second and the third joint) in its structure. This mechanism was designed to overcome the high forces that appear (payload of the robot is 1250 N) and high dynamics loads. The closed loop mechanism in the robot structure represents a restriction in the workspace, but due to the nature of the robot applications (operation of the robot in a small area of the workspace) this restriction on the workspace did not represent an obstacle, so it was not included in our models of direct and indirect kinematics for simulation.

Program equipment Catia V5 R17, which was used to design the ACMA robot, makes it possible to export the design model into the format VRML, which provides a wide range of options for further virtual simulations.

Fig. 8 presents the ACMA robot model in the VRML (Virtual Modelling Language) environment that enables indirect programming and possibility to control the pre-programmed trajectories.

The indirect programming runs under the user interface on the basis of MS Visual.NET [5] to [11]. The interface uses points of the desired trajectory for motion synthesis. The visualization of the simulation enables the operator to see whether the trajectory is in the expected shape and area of the workspace and if there are no singular points on the path [30] to [38].

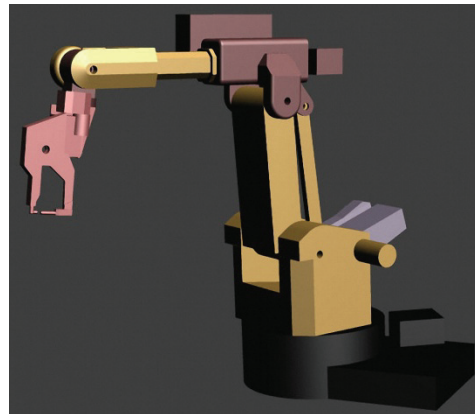


Fig. 8. VRML ACMA robot model

#### 4 CONCLUSION

In this paper an example of developing direct and indirect kinematic models in virtual space with reverse engineering is shown. The aim was to present the possibility to update older robot systems. With the virtual model indirect access programming is possible. Virtual simulation of a robotic cell or system gives us the opportunity for contemporary access to the design and modelling of robot applications also with robots that are out-of-use because of their out-of-date software or their user-unfriendly programming software.

Further work will include upgrading of the programming interface with new intelligent

algorithms for robot motion optimization for indirect robot programming in virtual space.

#### 5 REFERENCES

- [1] Balič, J., (2004) *Computer integration of manufacturing*, Faculty of Mechanical Engineering, University of Maribor, Maribor, p. 50-110. ISBN 86-435-0399-1
- [2] Balič, J., (2002) *CAD/CAM processes*, Faculty of Mechanical Engineering, University of Maribor, Maribor, p. 80-150.
- [3] Balič, J., (2004) *Intelligent processing systems*, Faculty of Mechanical Engineering, University of Maribor, Maribor, p. 199-211.
- [4] Perme, T., (2006) Industrial robotics, *IRT3000*, December 2006, p. 84-87.
- [5] SolidPro - RobotWorks, Langenau, Germany, <http://robotworks-eu.com>, 16.05.2008.
- [6] Balič, J., (2007) Intelligent CAD\CAM system for CNC programming-an overview. *Advances in Production Engineering % Management*, vol. 1, p. 13-21.
- [7] Shishir Bhat, B.N., (2008) Profits and reduce cycle time with manufacturing cells. *Advances in Production Engineering % Management*, vol. 3, no. 1, p. 17-26.
- [8] Čuš, F., Zuperl, U., Kiker, E. (2007) A model-based system for the dynamic adjustment of cutting parameters during a milling process, *Strojniški vestnik - Journal of Mechanical Engineering*, vol. 53, no. 9, p. 1-17.
- [9] IronCad, Atlanta, <http://www.ironcad.com>, 20.05.2008.
- [10] ABB Ltd, Zurich, Švica, <http://www.abb.com/>, 24.05.2008.
- [11] Rejc, J., Munih, M. (2005) Programs for indirect programming of industrial robots, Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, p. 2-5.
- [12] Zupan, B., Remškar, A. (2006) Direct capture of scanned objects in a 3D CAD program, *ib-CADdy*, d.o.o. Ljubljana, Toolmaking, p. 3-5.
- [13] Balič, J., Klančnik, S., Brezovnik, S., (2008) Feature extraction from CAD Model for milling strategy prediction, *Strojniški vestnik - Journal of Mechanical Engineering*, vol. 54, no. 4, p.301-307.
- [14] Klančnik, S., Balic, J., Planinsic, P. (2007) Obstacle detection with active laser triangulation. *Advances in Production Engineering & Management*, vol. 2, p. 79-90.
- [15] Coiffet, P., (1983) Modelling and control, robot technology vol. 1, Kogan Page, London, p. 39-53.
- [16] Vukobratović, M., (1990) Robotics basics, p. 35-80.
- [17] Bajd, T., (2004) Basics of robotics, University of Ljubljana, Ljubljana, p. 15-30.
- [18] Bajd, T. (2002) Robotics, University of Ljubljana, p. 10-34.
- [19] Safaric, R., Sinjur, S., Zalik, B., Parkin, R.M. (2003) Control of robot arm with virtual environment via the internet. Proc. I.E.E.E., Mar 2003, vol. 91, iss. 3, p. 422-429.
- [20] Safaric, Riko: Robotic I [[http://www.ro.feri.unimbi.si/predmeti/robotika\\_1/Welcome.html](http://www.ro.feri.unimbi.si/predmeti/robotika_1/Welcome.html)], Faculty of Electrical Engineering and Computer Science, 2007.
- [21] Foo, K.S., Gonzales, R.C., Lee, C.S.G. (1987) *ROBOTICS Control, Sensing, Vision and Intelligence*. International edition, McGraw Hill.
- [22] Craig, J., (2005) Introduction to robotics: mechanics and Control, Upper Saddle River: Pearson Education International, p. 30-32.
- [23] Product Management & Marketing Department at ABB Robotics Paint: Paint Manual, Bryne, October 2003.
- [24] Lenarčič, J., Bajd, T., (2003) Robotics mechanisms, University of Ljubljana, Publisher FE in FRI, Ljubljana, p. 55- 105.
- [25] Product Specification, Paint Automation (Publ.No.: 3HNT 00016-1 en Rev. 17); *ABB Flexible Automation AS*, Bryne, Norway, 1997-2003.
- [26] Shimon Y., Nof, *Handbook on Industrial Robotics*, Second Edition, John Wiley & Sons Inc., Canada 1999.
- [27] ABB Automation Technologies, Robotics. Foundry Automation from ABB, June 2003.
- [28] LaRoux K. Gillespie, Deburring and edge finishing handbook, Society of Manufacturing Engineers, USA, 1999.
- [29] Denavit, J., Hartenberg, R.,S. (1955) A kinematic notation for lower-pair mechanisms based on matrices. Trans ASME J. Appl. Mech, p. 215-221.



- [30] Hess, M. (2003) Höchste Zeit für den Roboter, Giesserei, *Die Zeitschrift der Deutschen Giesserei-vereinigungen*, p. 46 – 47.
- [31] Khatib, O. (1986) Real-time obstacle avoidance for manipulators and mobile robots, *J. of Robotic Systems*, Volume 5, p. 90-98.
- [32] Nemec, B., Zlajpah, L. (2000) Null velocity control with dynamically consistent pseudo-inverse, *Robotica*, 18, p. 513 - 518.
- [33] Yoshikawa, T. 1990, Foundations of robotics: analysis and control, MIT Press.
- [34] Yoshikawa, T. (1996) Basic optimization methods of redundant manipulators, *Laboratory Robotics and Automation*, vol. 8, no.1, p. 49-90.
- [35] Hogan, N. 1985, Impedance Control: An Approach to Manipulation: Part 1: Theory, Part 2: Implementation, Part 3: Applications, *Trans. of ASME J. of Dynamic Systems, Measurement, and Control*.
- [36] Dessault Systemes, France, <http://www.3ds.com/home>, 16.05.2008.
- [37] Koren, R., Robot cell for applying a protective layer; B.Sc. Thesis, Ljubljana, 2004.
- [38] Ranky, P., G., Ho, C., Y. (1985) Robot Modelling, *Control and applications with software*, IFS, Springer – Verlag, p. 41-193.