Proceedings in Manufacturing Systems, Volume 13, Issue 4, 2018, 189-194 ISSN 2067-9238

DEDICATED FORWARD KINEMATICS ALGORITHMS FOR SCARA AND PALLETIZING ROBOTS

Cezara COMAN[1], Mario IVAN[2,*], Cozmin CRISTOIU[3], Bogdan VERDETE[4]

[1) 2)] Lect., PhD, Machines and Manufacturing Systems Department, University "Politehnica" of Bucharest, Romania
[3), 4)] Assist. Prof., PhD Student, Machines and Manufacturing Systems Department, University "Politehnica" of Bucharest, Romania

Abstract: *This paper presents the development and validation of optimized mathematical algorithms for solving the forward kinematics problem in the case of two industrial robot architectures: SCARA and palletizing robots. While there are well established methods for describing the position and orientation of mechanism links with respect to a reference frame – in the case of industrial robots, the base frame – these algorithms, such as the Denavit-Hartenberg convention or the quaternion approach, are very flexible and applicable for all industrial robots with serial architecture. Thus, these algorithms do not necessarily provide the most efficient approach in all cases. Since there are certain kinematic particularities for each robot architecture, this article intends to exploit the specific structural properties of SCARA and palletizing robots in order to develop more efficient, dedicated algorithms for solving the forward kinematics equations. For each algorithm, the input parameters were considered to be the distance between axes and joint parameters, while the output parameters were considered to be the x, y and z coordinates of the tool center point with respect to the base frame, as well as a rotation matrix expressing the orientation of the tool center point frame with respect to the base frame of the robot.The algorithms were validated using the Catia V5 software by using the DMU Kinematics module to configure the robot kinematics.*

Key words: *forward kinematics, mathematical algorithm, SCARA robot, palletizing robot, kinematic validation.*

1. INTRODUCTION

In the world of trajectory generating mechanical systems, in general, and especially in the field of industrial robotics, the ability of calculating and, from a mathematical point of view, modelling the trajectory is essential. In order to generate particular path segments, such as linear or circular, an articulated arm robot with six degrees of freedom, for example – which is one of the most kinematic flexible architecture – must use all six axes with certain speeds. Furthermore, although an industrial robot will show on the teach pendant screen the values of the tool center point (TCP) position and end-effector orientation in real time, it has no mean of read in these values directly. All the above are calculated by the controller using mathematical algorithms.

The most elementary of all algorithms integrated in controller programming are the forward kinematics and inverse kinematic problems [1]. The statement of the forward kinematic problem is that, knowing the robot link dimensions and all joint values, the TCP position and end-effector orientation can be calculated. The inverse kinematic problem requires the calculation of all joint values, knowing the TCP position, end-effector orientation and link dimensions. While the forward kinematics problem is generally more straightforward and yields an unique result, an inverse kinematics model is generally non-linear and yields multiple results for the same input values, which correspond to multiple robot configurations for the same tool position and orientation. It should be noted, however, that these considerations are applicable for serial robot architectures, while for parallel structures the nature of the algorithms is different [2].

At this moment, there are well established general algorithms for solving the forward kinematic model. The most documented and applied model is the Denavit-Hartenberg convention. This algorithm is based on matrix calculations to model the transition from one link to another, up to the TCP frame and can be applied to any serial link manipulator, including industrial robot. Another approach is represented by expressing the transitions between robot links using quaternions. Both of these algorithms, while very versatile, have the inconvenience of including in each transformation the parameters required for calculating the new orientation, even if there is no orientation modifier. This is the case for prismatic joint, which introduce only position transformations. Furthermore, the matrix-based calculations require a high number of mathematical operations, while, for an industrial robot with less than five rotary joints, the x, y and z coordinates of the TCP with respect to the base frame can be expressed through first degree equations.

Taking the above aspects into considerations, there are certain industrial robot architectures for which dedicated, more efficient forward kinematics algorithm

*Corresponding author: 313 Splaiul Independentei, Bucharest, Romania, UPB – Faculty of IMST, MSP Department;
Tel.: + 40 21 402 91 00;
E-mail addresses: *andrei.mario@yahoo.com*.

$$
{}^{n-1}T_n = \left[\begin{array}{ccc|c} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & r_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & r_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} & R & & T \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
$$

Fig. 1. Denavit-Hartenberg coordinate transformation matrix.

can be developed. The research approached the SCARA and palletizing structures.

2. STATE OF THE ART

In order to describe the general context in which the present work is placed, the state of the art in the field of forward kinematics will be illustrated.

The main approach for solving the forward kinematics problem is the Denavit-Hartenberg convention. This algorithm is based on the principle of describing the position and orientation of each link with respect to previous link, starting from the base of the robot. In order to achieve this, the algorithm assigns a frame to each link, with the base frame being attached to the base of the robot. The position and orientation of each link with respect to the previous link is expressed using a coordinate transformation matrix, which has the form illustrated in Fig. 1 [3]. The R submatrix describes the orientation modification, and the T submatrix describes the position modification. Each coordinate transformation matrix is based on four parameters, d, θ, r and α, which describe the translations and rotations that the previous frame should perform to come into the same position and orientation as the new frame.

The advantages of the Denavit-Hartenberg approach are the relative ease of implementation in programming, the applicability to multiple robot serial architectures and relatively few transformation parameters. Still, for all its reliability, the algorithm has its own drawbacks. Being a general convention, it lacks optimization for particular robot architectures. Also, each transformation has orientation and position components, even for prismatic joints – which do not introduce orientation modifiers. Thus, for certain robot models with serial architecture, more efficient, more suitable algorithms for solving the forward kinematics problem can be developed.

3. SCARA ROBOT FORWARD KINEMATICS

The development of the forward kinematics algorithm for SCARA robots is based on the specific kinematic diagram of this architecture. For demonstrating the equations, aRRT (rotation-rotation-translation) SCARA robot is used as an example – the Adept Cobra 450, illustrated in Fig. 2 [4]. The workspace and dimensions of the robot are illustrated in Fig. 3. The kinematic diagram of the robot is illustrated in Fig. 4, together with link dimensions. The diagram is configured with all axes in “0” position. In order to correctly configure the robot joint parameters, each axis travel limit is specified below:

- Joint 1 – ±125^0.
- Joint 2 – ±145^0.
- Joint 3 – 200 mm.
- Joint 4 – ±360^0.

By analyzing the kinematic diagram of the robot, several specific features can be observed:

- The robot has four axes, three of which are rotary axes.
  The axes of all three rotary joints are parallel along the *Z* axis of robot’s base coordinate system.
- The orientation of the end-effector is influenced only by the three rotary joints (Joint 1, Joint 2 and Joint 4). These axes being parallel to the *Z* axis of the base coordinate system, the orientation of the end-effector

Fig. 2. Adept Cobra 450.

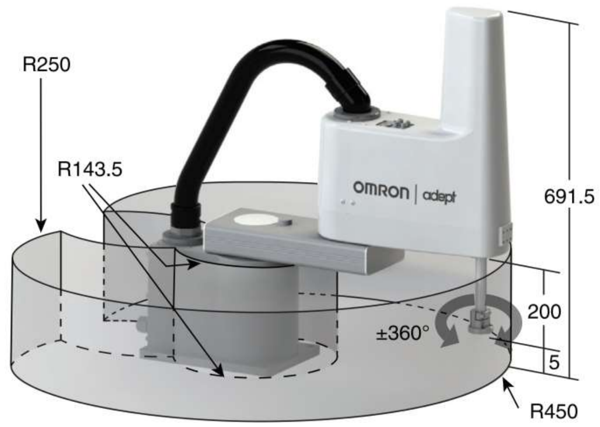

Fig. 3. Adept Cobra 450 workspace and dimensions.

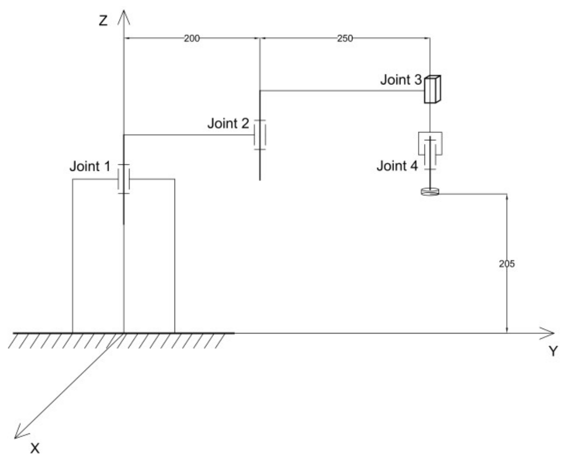

Fig. 4. Adept Cobra 450 kinematic diagram.

- is modified only around the *Z* axis of the tool center point frame.
- The x and y coordinates of the TCP are influenced only by Joint 1 and Joint 2.
- The z coordinate of the TCP is influenced only by the prismatic joint (Joint 3).

Taking into account the above aspects, it can be concluded that the position and orientation coordinates of the end-effector can be determined by dividing the problem into several stages:

- Stage 1 – calculation of the x and y coordinates of the TCP.
- Stage 2 – calculation of the z coordinate of the TCP.
- Stage 3 – calculation of the orientation of the TCP.

In order to solve the Stage 1 problem, the aspect can be reduced to analyzing an articulated, rotation-rotation type of planar mechanism. The two rotary joints of the mechanism correspond to Joint 1 and Joint 2 of the SCARA robot. These are the joints that influence the x and y coordinates of the TCP. The link between joints and the final link have the lengths notes as l_1 and l_2 respectively, in order to obtain the necessary equations. The final link ends with the TCP. This concept is illustrated in Fig. 5. The view is oriented as a normal view on the *XY* plane. The rotation parameter of Joint 1 is α and the rotation parameter of Joint 2 is β.

The first step is calculating the coordinates for the center of Joint 2:

$$x_{J2} = l_1\cos(\alpha) \quad (1)$$

$$y_{J2} = l_1\sin(\alpha) \quad (2)$$

We shall consider another frame – X_1Y_1 – with the origin in the center of Joint 2. The angle between the second link of the mechanism and the X_1 axis is α-β. Thus, the coordinates of the TCP with respect to the X_1Y_1 frame are:

$$x'_{TCP} = l_2\cos(\alpha - \beta) \quad (3)$$

$$y'_{TCP} = l_2\sin(\alpha - \beta) \quad (4)$$

From the previous equations, the coordinates x and y of the TCP with respect to the *XYZ* base frame are:

$$x_{TCP} = x_{J2} + x'_{TCP} = l_1\cos(\alpha) + l_2\cos(\alpha - \beta) \quad (5)$$

$$y_{TCP} = y_{J2} + y'_{TCP} = l_1\sin(\alpha) + l_2\sin(\alpha - \beta) \quad (6)$$

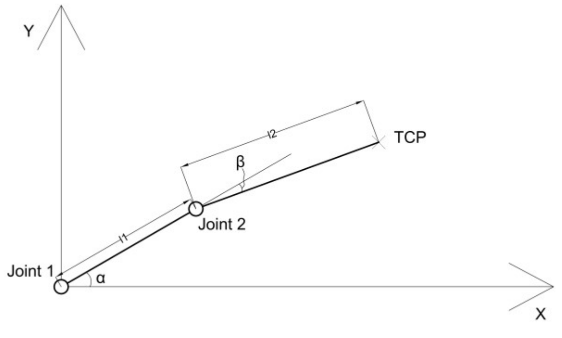

Fig. 5. Diagram for calculating the *x* and *y* coordinates of the TCP.

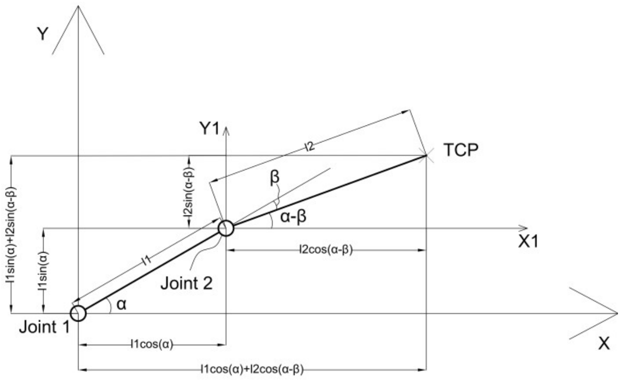

Fig. 6. Calculation of the x and y coordinates of the TCP.

The above equations are graphically represented in Fig. 6.

For Stage 2, it can be observed from Fig. 4 that, with Joint 3 in the "0" position, the TCP is 205 mm above XY plane of the base frame. This is the z coordinate of the TCP when the robot has all joints in the "0" position and can be influenced only by Joint 3. Thus, when Joint 3 causes the TCP to lower, the parameter value of the prismatic joint is subtracted from the 205 mm value corresponding to the "0" position, thus giving the expression for the z coordinate of the TCP:

$$z_{TCP} = 205 - a \quad (7)$$

where *a* is the travel of Joint 3. This expression is graphically represented in Fig. 7.

For Stage 3, the orientation of the end-effector must be calculated. Because all three rotary joints that influence the orientation of the end-effector are parallel to the Z axis of the base frame, it can be expressed as a rotation matrix with an angle composed as the sum of the three parameters of Joint 1, Joint 2 and Joint 4 – α, β and γ respectively.

$$R_{TCP} = R_z(\alpha) \cdot R_z(\beta) \cdot R_z(\gamma) = R_z(\alpha + \beta + \gamma) \quad (8)$$

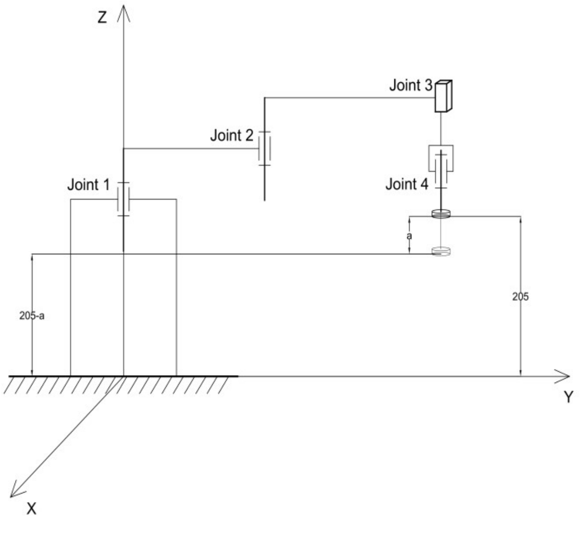

Fig. 7. Calculation of the z coordinate of the TCP.

$$R_z(\alpha+\beta+\gamma) = \begin{pmatrix} \cos(\alpha+\beta+\gamma) & -\sin(\alpha+\beta+\gamma) & 0 \\ \sin(\alpha+\beta+\gamma) & \cos(\alpha+\beta+\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (9)$$

4. PALLETIZING ROBOT FORWARD KINEMATICS

The development of the forward kinematics algorithm for palletizing robots is based on the specific kinematic diagram of this architecture. For demonstrating the equations, a Kuka KR 120 R3200 PA-HOpalletizing robot is used as an example, illustrated in Fig. 8 [5]. The workspace and dimensions of the robot are illustrated in Fig. 9. The kinematic diagram of the robot is illustrated in Fig. 10, together with link dimensions. It should be noted that Joint 3' is automatically synchronized in order to keep the flange axis always in vertical position. The diagram is configured with all axes in "0" position. In order to correctly configure the robot joint parameters, each axis travel limit is specified below:

- Joint 1 – $\pm 185^0$.
- Joint 2 – -140^0 / -5^0.
- Joint 3 – 0^0 / 155^0.
- Joint 4 – $\pm 350^0$.

Let P be the notation for the TCP and P_1 be the notation for the projection of P on the XY plane of the base frame. In order to determine the x, y and z coordinates of P, the kinematic structure of the robot illustrated in Fig. 10 must first be projected onto the ZY plane, as illustrated in Fig. 11. Joint 1 was ignored in this representation, as well as Joint 4 (because Joint 4 has no influence over TCP position).

Fig. 8. KukaKR 120 R3200 PA-HO.

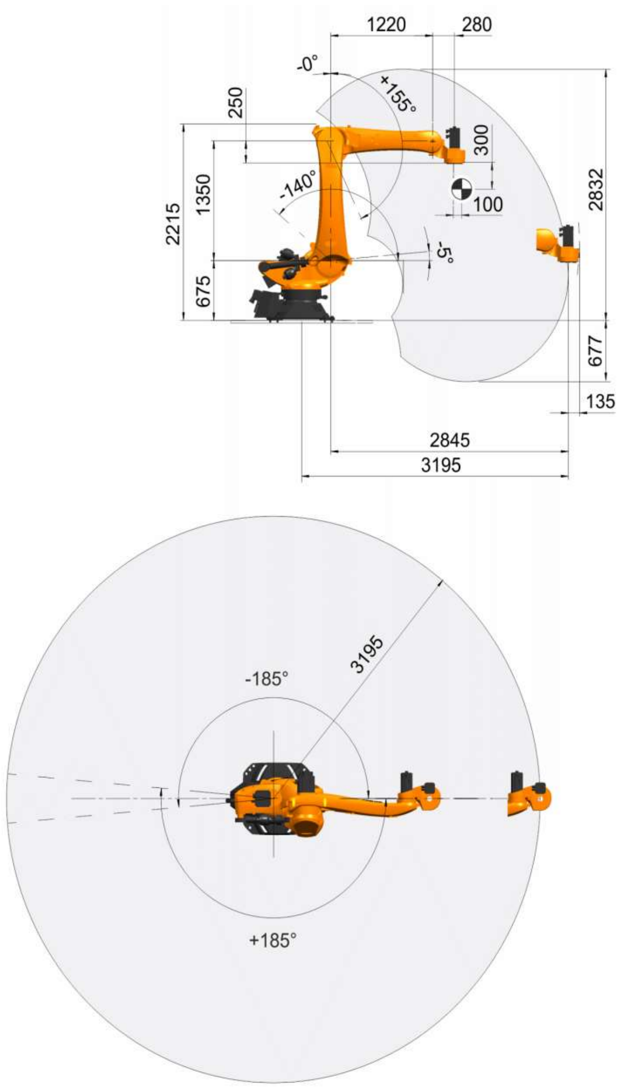

Fig. 9. KukaKR 120 R3200 PA-HO workspace and dimensions

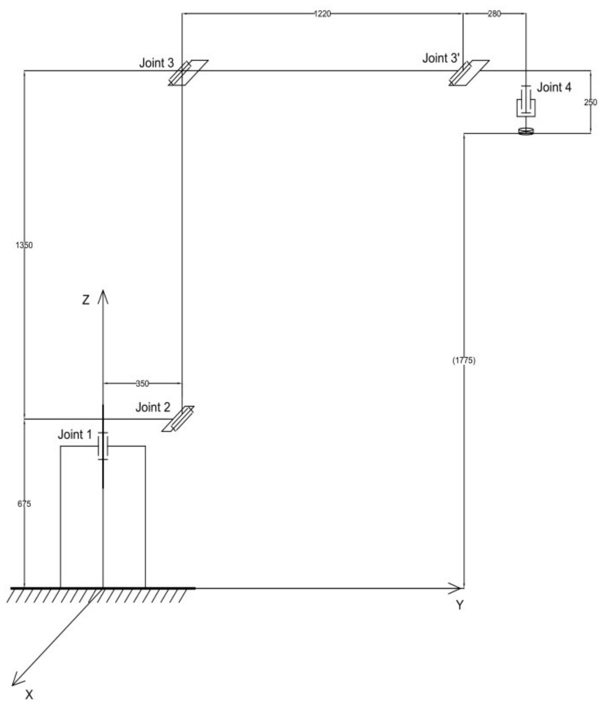

Fig. 10. KukaKR 120 R3200 PA-HO kinematic diagram.

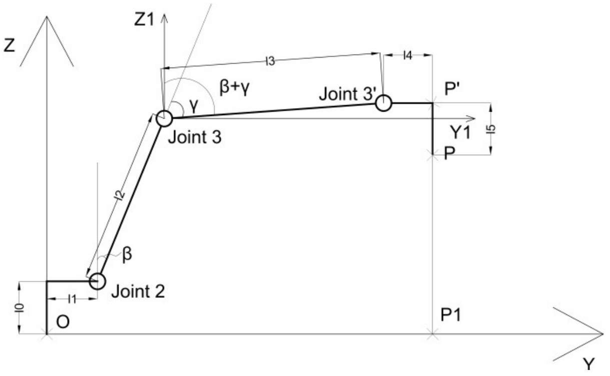

Fig. 11. KukaKR 120 R3200 PA-HO kinematic diagram projection onto ZY plane of the base frame.

It should be noted that, in Fig. 11, due to the kinematic structure of the robot, the P’P segment is always parallel to the Z axis, for any robot configuration.

From the kinematic diagram in Fig., 11, the OP_1 length can be calculated. However, OP_1 does not represent the y coordinate of the TCP, as P_1 lies on the Y axis only when the robot has Joint 1 in “0” position.

First, the coordinates for the center of Joint 3 must be calculated:

$$y_{J3} = l_1 + l_2\sin(\beta) \quad (10)$$
$$z_{J3} = l_0 + l_2\cos(\beta) \quad (11)$$

For the second step, the coordinates of P’ in the Z_1Y_1 frame must be calculated:

$$y_{P'} = l_3\sin(\beta+\gamma) \quad (12)$$
$$z_{P'} = l_3\cos(\beta+\gamma) \quad (13)$$

Then,

$$OP_1 = l_1 - l_2\sin(\beta) + l_3\sin(\beta+\gamma) + l_4 \quad (14)$$
$$z_P = l_0 + l_2\cos(\beta) + l_3\cos(-\beta+\gamma) - l_5 \quad (15)$$

Considering the normal view on the XY plane of the base frame (illustrated in Fig. 12) in which only Joint 1 and the OP_1 segment are represented, the coordinates x and y of the TCP can be calculated:

$$x_P = OP_1\cos(\alpha) = (l_1 - l_2\sin(\beta) + l_3\sin(-\beta+\gamma) + l_4)\cdot(-\sin(\alpha)) \quad (16)$$
$$y_P = OP_1\sin(\alpha) = (l_1 - l_2\sin(\beta) + l_3\sin(-\beta+\gamma) + l_4)\cdot\cos(\alpha) \quad (17)$$

The orientation of the end-effector is only influenced by Joint 1 and Joint 4. The axes for these joints are both parallel to the Z axis of the base frame, thus the orientation can be expressed through a rotation matrix around the Z axis an angle composed as the sum of α and θ – parameters of Joint 1 and Joint 4 respectively.

$$R_{TCP} = R_z(\alpha)\cdot R_z(\gamma) = R_z(\alpha+\beta+\gamma) \quad (18)$$

$$R_z(\alpha+\theta) = \begin{pmatrix} \cos(\alpha+\theta) & -\sin(\alpha+\theta) & 0 \\ \sin(\alpha+\theta) & \cos(\alpha+\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (19)$$

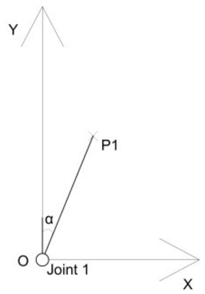

Fig. 12. OP_1 segment on the XY plane of the base frame.

5. VALIDATION OF FORWARD KINEMATICS ALGORITHM

The forward kinematics algorithms for the SCARA and palletizing robots were validated by implementing the position equations and rotation matrices inside the MathCAD software. The software was used to calculate the results of the equations by assigning values to the input parameters. Furthermore, the 3D virtual model of the robots chosen as case studies were imported in Catia V5 where the kinematic structures were configured using the DMU Kinematics model. The compass was then used to read the position and orientation values for the TCP frame. The validation was made through comparison between the results provided by the two applications. For each architecture the validation was made using four sets of parameters. Sample validation images are shown below for one set of parameters. The validation parameter sets for the SCARA robot are presented in Table 1. The sample validation images for the SCARA robot are illustrated in Fig. 13 for CATIA V5 implementation and in Fig. 14 for MathCAD implementation. The validation parameter sets for the palletizing robot are presented in Table 2. The sample validation images for the palletizing robot are illustrated in Fig. 15 for CATIA V5 implementation and in Fig. 16 for MathCAD implementation.

Table 1

Validation joint parameters for SCARA robot

	1st validation	2nd validation	3rd validation	4th validation
Joint 1	-35^0	45^0	-115^0	105^0
Joint 2	-45^0	65^0	-125^0	120^0
Joint 3	0 mm	30 mm	80 mm	160 mm
Joint 4	-45^0	35^0	-170^0	245^0

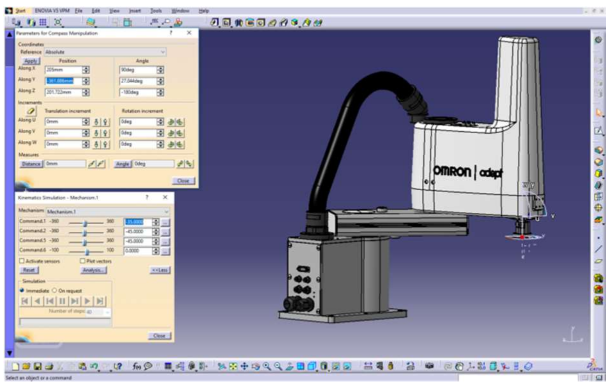

Fig. 13. Sample validation images for the SCARA robot – CATIA V5.

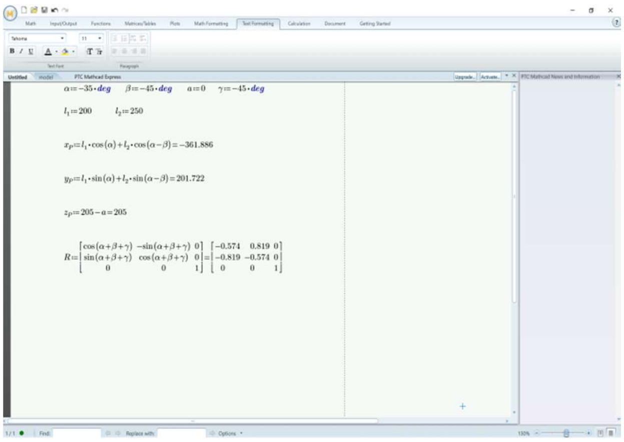

Fig. 14. Sample validation images for the SCARA robot – MathCAD.

Table 2

Validation joint parameters for palletizing robot

	1^{st} validation	**2^{nd} validation**	**3^{rd} validation**	**4^{th} validation**
Joint 1	-65^0	85^0	-15^0	-105^0
Joint 2	-15^0	-65^0	-95^0	-120^0
Joint 3	30^0	60^0	100^0	130^0
Joint 4	-45^0	75^0	-10^0	-145^0

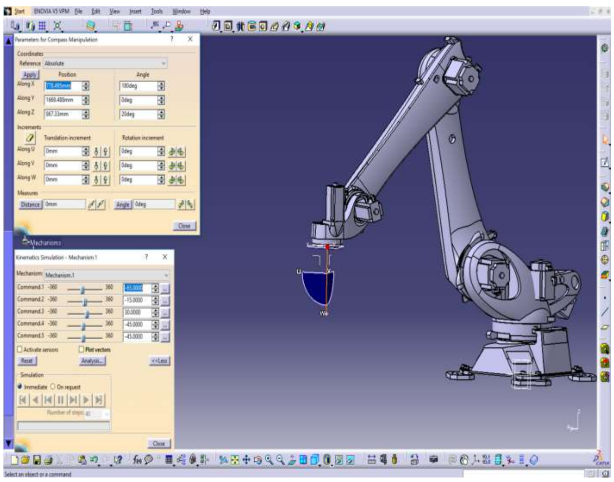

Fig. 15. Sample validation images for the palletizing robot – CATIA V5.

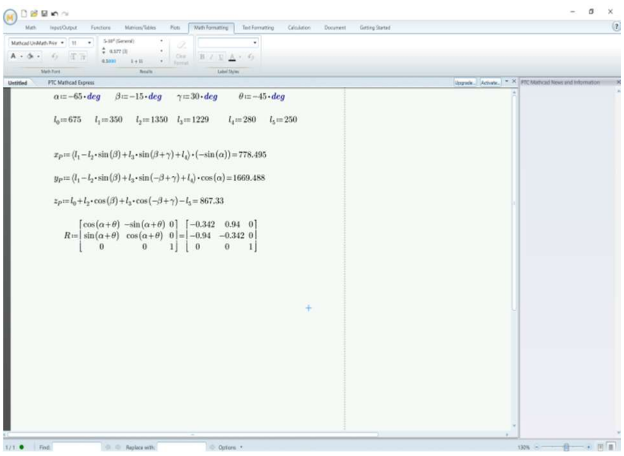

Fig. 16. Sample validation images for the palletizing robot – MathCAD.

6. CONCLUSIONS

Currently, the most used forward kinematics solving algorithm is the Denavit-Hartenberg convention. This formalism has certain advantages, one of the most important being that it can model the transformation between link positions and orientation by using only four parameters instead of six. Also, Denavit-Hartenberg is a flexible algorithm which, in theory, can be used for any serial link manipulator and thus, for industrial robots with serial architectures. However, in practice, considering the major robot architectures that are widely used at this moment in industry, the Denavit-Hartenberg algorithm is suitable and efficient only for articulated arm robots with five or more axes. Certain robot architectures however have kinematic particularities that can be used to develop dedicated, optimized, more efficient algorithms.

The algorithms presented in this paper are based on geometric approaches for solving the forward kinematics problem. The development of these approaches are based on certain kinematics particularities of the analyzed robots. First of all, rotary joints that have parallel axes determine orientation transformations that can be described using a single, elementary rotation matrix. Besides, all rotary joints with parallel axes in a kinematic robot structure form a planar articulated mechanism, which can be easily modeled. Furthermore, prismatic joints do not introduce orientation transformations.

Starting from the above considerations, optimized and dedicated forward kinematics algorithms were developed for SCARA and palletizing architectures. In order to work with real robot models, and not just configure a purely theoretical structure, a case study was used for each architecture – Adept Cobra 450 for the SCARA architecture (with a rotation-rotation-translation workspace mechanism structure) and Kuka KR 120 R3200 PA-HO for the palletizing architecture. The specific kinematic features for each robot were taken into consideration. The forward kinematics problem was then divided into several easier to solve problems by projecting the kinematic structures of the robots onto various planes of the reference frame. The position of the TCP was expresses, in each case through three equations, one for each of the x, y and z coordinates. Also, for each analyzed architecture, the orientation of the TCP frame was expressed through a rotation matrix formed on the basis of rotary joint parameters corresponding to the axes that are parallel to the Z axis of the base frame.

REFERENCES

[1] B.Siciliano, L.Sciavicco, L.Villani, G.Oriolo, *RoboticsModelling, Planning and Control*, Springer Science & Business Media, 2009

[2] A.Nicolescu, *RobotiIndustriali*, EDP, 2005.

[3] A. Ivan, *Research regarding optimization of industrial robots for machining applications,* doctorate thesis, Politehnica University of Bucharest, 2011

[4] https://www.adept.com/, accessed: 2016-05-12

[5] https://www.kuka.com/en-de, accessed: 2016-05-16