Proceedings in Manufacturing Systems
Proceedings in Manufacturing Systems, Volume 12, Issue 4, 2017, 145-153 ISSN 2067-9238

VALIDATION OF FORWARD GEOMETRIC MODELS FOR ABB ROBOTS USING VIRTUAL MODELS AND THE SOFTWARE APPLICATIONS CATIA AND ABB ROBOT STUDIO

Cozmin CRISTOIU[1,*], Adrian NICOLESCU[2]

[1)] Assist. Prof., PhD student, Machines and Production Systems Department, University "Politehnica" of Bucharest, Romania
[2)] Professor, PhD, Machines and Production Systems Department, University "Politehnica" of Bucharest, Romania

Abstract: *This paper presents the working steps for the validation of the forward geometric models of ABB industrial robots (IR) using their virtual models (CAD models) along with CATIA and ABB RobotStudio software applications. Applying this method, one can obtain on the basis of the virtual model a series of results regarding the position of the characteristic point of the robot according to joint positions, results that can be compared with the results of the mathematical model analytically calculated for different configurations. By comparing these results, one can determine the correctness of the forward geometric models.*

Key words: *industrial robots, new modeling approach, forward geometric model, cross validation methods.*

1. INTRODUCTION

As part of the work in progress for PhD thesis "Research on thermal behavior of industrial robots and improvement of their performances by compensating thermal errors", it was necessary to develop custom forward geometric models for three robot models: ABB IRB 120, ABB IRB 140 and ABB IRB 460. The forward geometric models were developed by following an own approach was presented in [1 and 2] resulting in particular geometric models, different from the models found in the literature. For these models first of all validation was required. The validation of these models was done in two stages:

1. comparing the results of these particular models with the results of some other existing geometric models from related literature;
2. comparing the results of these particular models with the results obtained using virtual models and some software applications. Further, this paper presents a simple and quick solution for validation of forward geometric models using CATIA and ABB RobotStudio applications and virtual robot models.

2. STUDIED ROBOTS

In developing the forward geometric model of an RI, it usually starts from a simple kinematic wireframe based on some constructive parameters. After the realization of the wireframe with the symbolic representation of the elements and with the assignment of the reference frames describing the relative position of these elements, the mathematical modeling is made having as support this simple kinematic scheme. Some constructive and functional parameters taken from the datasheets of the robots are given further.

All three robots have articulated arm architectures with the following features: ABB IRB 120 – a 6-degree robot with a bilateral symmetry structure; ABB IRB 140 robot with 6 degrees of freedom with asymmetrical structure and ABB IRB 460 robot with 5 degrees of freedom (only four numerically controlled axes) and with a closed cinematic chain structure.

Figure 1 and Table 1 present the technical specifications of ABB IRB 120 industrial robot.

Table 1

Technical specifications of ABB IRB 120

Main applications	**Physical**	
"pick and place"	Base dim. 180 × 180 mm	
Specifications	Weight 25 kg	
Max. payload 3 kg	**Performances (ISO 9283)**	
Max reach 0.58 m	Repeatability 0.01 mm	
Axes 6	Accuracy 0.02 mm	
Protection IP30	**Axis movement**	
Mounting – any angle	A1 ±165°,250°/s	A5 ±120°, 320°/s
Controller IRC5	A2 ±110°,250°/s	A6±400°, 420°/s
Powe supply 200–600 V, 50/60 Hz	A3 +70°,- 110°,250°/s	
Consumed power 0.25 kW	A4 ±160°,320°/s	

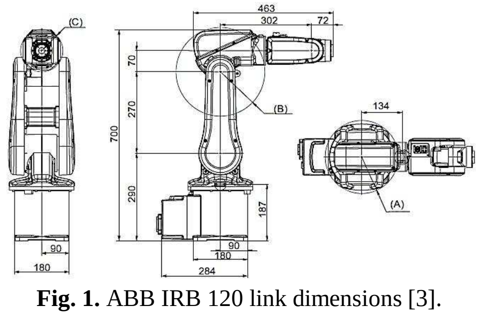


Fig. 1. ABB IRB 120 link dimensions [3].

* Corresponding author: Splaiul Independenței 313, district 6, 060042, Bucharest, Romania,
Tel.: 0040 766714482,
E-mail addresses: *cozmin.cristoiu@gmail,.com* (C. Critstoiu), *afnicolescu@yahoo.com* (A. Nicolescu)

Table 2

Technical specifications of ABB IRB 140

Main applications	**Physical**	
part extraction of die-casting	Base dim. 400 × 450 mm	
Specifications	Weight 98 kg	
Max. payload 6 kg	**Performances (ISO 9283)**	
Max. reach 0.81 m	Repeatability 0.03 mm	
Axes 6	Accuracy 0.02 mm	
Protection IP67	**Axis movement**	
Mounting: ground/suspended	A1 ±360°,200°/s	A5 ±240°, 360°/s
Controller IRC5	A2 ±200°,200°/s	A6 ±800°, 450 °/s
P. Supply 200-600V, 50/60 Hz	A3 ±280°,260°/s	
Consumed power 0.44 kW	A4 ± 400°,360°/s	

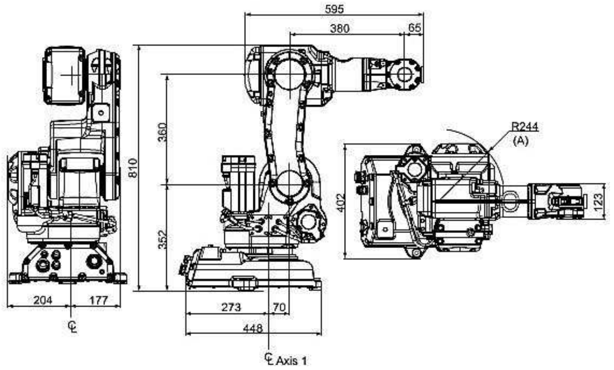


Fig. 2. ABB IRB 140 link dimensions [4].

Table 3

Technical specifications of ABB IRB 460

Main applications	**Physical**
Palletizing, material handling	Base dim. 1007 x 720 mm
Specifications	Weight 925 kg
Max. payload 110 kg	**Performances (ISO 9283)**
Max. reach 2.40 m	Repeatability 0.2 mm
Axes 4	Accuracy 0.3 mm
IP67	**Axis movement**
Mounting: on ground	A1 ±165°,145°/s
Controller IRC5	A2 +85°,-40°,110°/s
P. Supply 200-600V,50/60Hz	A3 +120°,-20°,120°/s
Consumed power 4.31 kW	A4 ±300°,400°/s

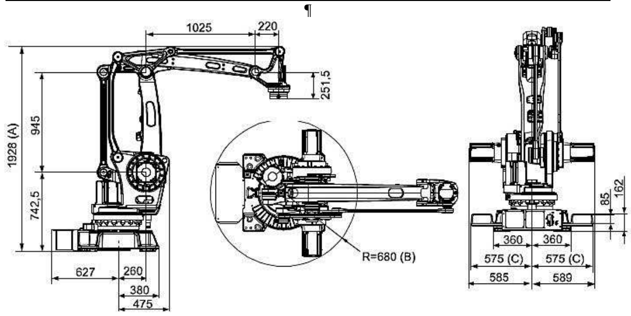


Fig. 3. ABB IRB 460 link dimensions [5].

Figure 2 and Table 2 present the technical specifications of ABB IRB 140 industrial robot.

Figure 3 and Table 3 present the technical specifications of ABB IRB 460 industrial robot.

3. VALIDATION USING CATIA

For all three robots, the virtual models were downloaded from the manufacturer's CAD database and imported into CATIA in turn. Using specific commands from DMU Kinematics workbench, joints are defined between the robot segments with the possibility of changing their position. Figure 4 exemplifies how to define a joint (example joint 3 of IRB 120 model) in the DMU Kinematics workbench from CATIA. Further, Figure 5 exemplifies how using the "Simulate with commands" command one can access the control window for joint positions in order to modify the robot configuration. Similarly, these things were done for all three robot models.

For mathematical computation, the transformation matrices and their multiplication used to determine the relative position of the robot elements and finally the position of the characteristic points were implemented in excel computational files. To verify the calculations, in the parameters table from Excel spreadsheets, the dimensions parameters (taken from the technical data sheet) should be set followed by the setting of the same values for joint angles as the values to be set on the virtual model "Simulate with commands" panel. In Tables 4 and 5 and images it is presented in parallel: positions measured on the virtual model with the help of

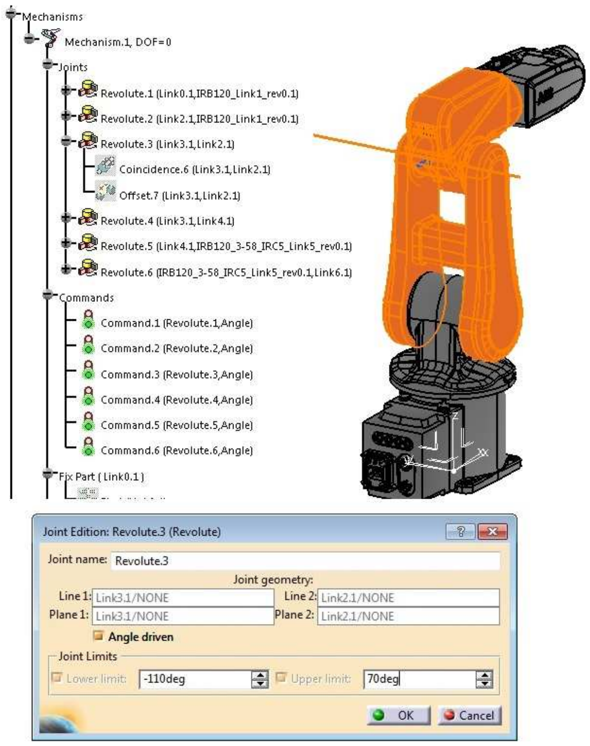


Fig. 4. Example of joint definition CATIA DMU Kinematics for IRB 120.

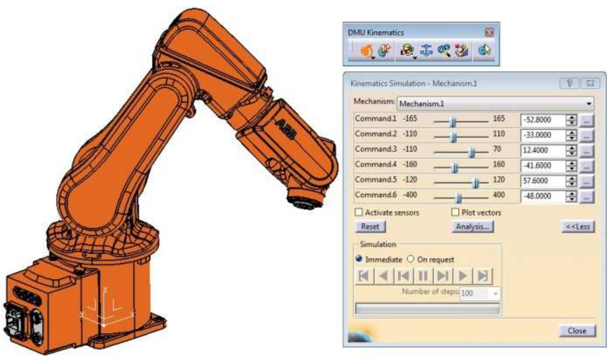


Fig. 5. Use of “Simulate with commands” panel to change robot configuration.

some measuring tags, the parameters table and the final computed matrices (these are including positions and the orientations of the characteristic points). These are presented for two robot positions: the "home" position and a random position. In Table 5, from the Excel file, the general form of parameter tables and analytically computed results contained in the final matrix obtained by successively multiplying the transformation matrices for each joint are extracted and presented.

Table 4

Measured position on virtual model in CATIA

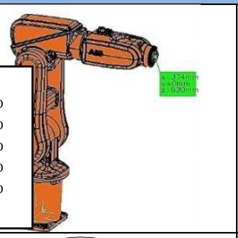
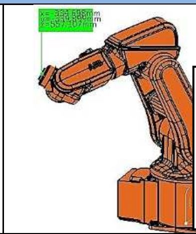
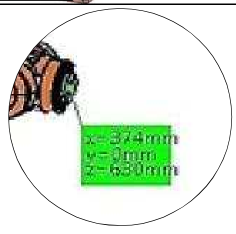
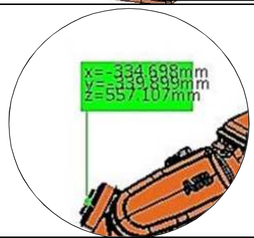
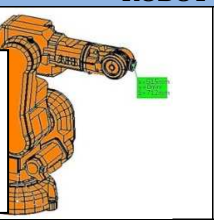
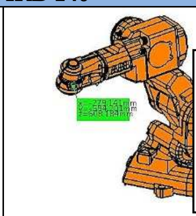
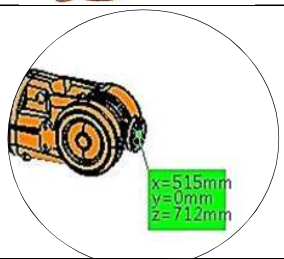
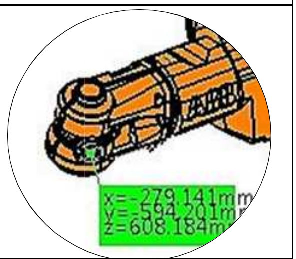
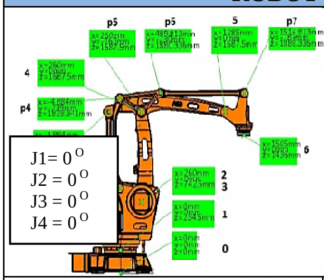
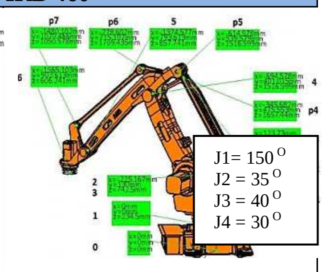
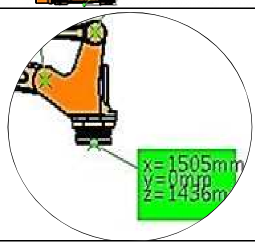
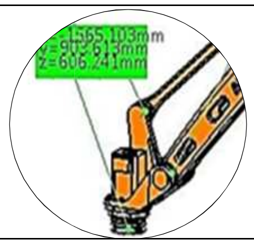
Home pose	Random pose
ROBOT IRB 120	
 J1= 0°, J2 = 0°, J3 = 0°, J4 = 0°, J5 = 0°, J6 = 0°	 J1= -130°, J2 = 25°, J3 = -10°, J4 = 40°, J5 = -55°, J6 = -5°
	
ROBOT IRB 140	
 J1= 0°, J2 = 0°, J3 = 0°, J4 = 0°, J5 = 0°, J6 = 0°	 J1= -120°, J2 = 30°, J3 = -25°, J4 = 70°, J5 = 65°, J6 = 15°
	
ROBOT IRB 460	
 J1= 0°, J2 = 0°, J3 = 0°, J4 = 0°	 J1= 150°, J2 = 35°, J3 = 40°, J4 = 30°
	

Table 5

Computed positions within excel file

ROBOT IRB 120

Joint rotation angles		Translations	Pe X	Pe axa Y	Pe axa Z
Θ_0	0	Sis. Ref 0	0	0	0
Θ_1	0	Sis. Ref 1	0	0	0
Θ_2	0	Sis. Ref 2	0	0	290
Θ_3	0	Sis. Ref 3	0	0	270
Θ_4	0	Sis. Ref 4	134	0	70
Θ_5	0	Sis. Ref 5	168	0	0
Θ_6	0	Sis. Ref 6	72	0	0

		T06 = T01*T12*T23*T34*T45*T56			
Θ_0	0				
Θ_1	0	1	0	0	374
Θ_2	0	0	1	0	0
Θ_3	0	0	0	1	630
Θ_4	0	0	0	0	1
Θ_5	0				
Θ_6	0				

		T06 = T01*T12*T23*T34*T45*T56			
Θ_0	0				
Θ_1	-130	-0.86387	0.464719	0.194312	-334.698
Θ_2	25	-0.21037	-0.68338	0.699097	-339.899
Θ_3	-10	0.457673	0.563054	0.688117	557.107
Θ_4	40	0	0	0	1
Θ_5	-55				
Θ_6	-5				

ROBOT IRB 140

Joint rotation angles		Translations	Pe X	Pe axa Y	Pe axa Z
Θ_0	0	Sis. Ref 0	0	0	0
Θ_1	0	Sis. Ref 1	0	0	0
Θ_2	0	Sis. Ref 2	70	0	352
Θ_3	0	Sis. Ref 3	0	0	360
Θ_4	0	Sis. Ref 4	254	0	0
Θ_5	0	Sis. Ref 5	126	0	0
Θ_6	0	Sis. Ref 6	65	0	0

		T06 = T01*T12*T23*T34*T45*T56			
Θ_0	0				
Θ_1	0	1	0	0	515
Θ_2	0	0	1	0	0
Θ_3	0	0	0	1	712
Θ_4	0	0	0	0	1
Θ_5	0				
Θ_6	0				

		T06 = T01*T12*T23*T34*T45*T56			
Θ_0	0				
Θ_1	-120	0.540554	0.039068	-0.8404	-279.141
Θ_2	30	-0.76703	-0.38749	-0.51138	-594.201
Θ_3	-25	-0.34563	0.921044	-0.1795	608.184
Θ_4	70	0	0	0	1
Θ_5	65				
Θ_6	15				

ROBOT IRB 460

Ref. axis translations for MAJOR joints	Pe X	Pe axa Y	Pe axa Z	Ref. axis translations for PASSIVE joints	Pe X	Pe axa Y	Pe axa Z
Sis. Ref 0	0	0	0	Sis. Ref c.p 1	-400	0	0
Sis. Ref 1	0	0	234.5	Sis. Ref c.p 2	0	0	945
Sis. Ref 2	260	0	508	Sis. Ref c.p 3	-246.884	0	140.841
Sis. Ref 3	260	0	508	Sis. Ref c.p 4	0	0	945
Sis. Ref 4	0	0	945	Sis. Ref c.p 5	264.884	0	-140.841
Sis. Ref 5	1025	0	0	Sis. Ref c.p 6	229.813	0	192.836
Sis. Ref 6	220	0	-251.5	Sis. Ref c.p 7	1025	0	0

Rotation angles for active joints		Rotation angles for passive joints		T01*T12*T24*T45*T5_reset*T56			
Θ_0	0	Θ_p1	0				
Θ_1	0	Θ_p2	0				
Θ_2	0	Θ_p3	0	0.866	-0.5	0	1505
Θ_3	0	Θ_p4	0	0.5	0.866	0	0
Θ_6	0	Θ_p5	0	0	0	1	1436
		Θ_p6	0	0	0	0	1
		Θ_p7	0				

Rotation angles for active joints		Rotation angles for passive joints		T01*T12*T24*T45*T5_reset*T56			
Θ_0	0	Θ_p1	-5				
Θ_1	150	Θ_p2	40	-1	-6E-17	6E-17	-1565.10301
Θ_2	35	Θ_p3	35	-6E-17	-1	-3E-17	903.6126433
Θ_3	40	Θ_p4	40	0	0	1	606.2413819
Θ_4	30	Θ_p5	-35	0	0	0	1
		Θ_p6	40				
		Θ_p7	0				

By comparing the measured values on the virtual model and the mathematically calculated values it can be observed that they are the same. This suggests that the forward geometric model for which the calculations were made is correct.

3. VALIDATION USING ABB ROBOT STUDIO

ABB RobotStudio is the official software application for off-line programming and simulation of ABB robots and robotic applications. The virtual robot models are already in the database of this application, are implicitly assembled and functional and have a virtual controller that is identical to the robot's real controller. ABB RobotStudio programs being directly loaded on the robot's real controller through a data cable. ABB RobotStudio is a very complex application and cannot be presented in just a few images with its many modules, working windows and functionalities. Figure 6 depicts a screen capture during the programming of a robot for a palletizing application, a capture which includes the simulation window, the signal control panel, the Rapid language text programming window and few elements from the command ribbon.

In ABB RobotStudio there are two main programming methods. For example, to record some positions and configurations of the robot, you can use the virtual teach-pendant first (identical to the real one) and control the robot in desired configurations and positions by using the control buttons and stick. Once you have reached the desired position, it can be recorded. It is possible to record a series of points and then generate a trajectory through these points. Also the teach-pendant can be used to programs using textual programming in Rapid language, of course being much difficult to write on the teach-pendant in comparison with the specialized editor from the RobotStudio application. Of course the real teach-pendant and the real robot could be used if available. Thus, if working with the real robot, a very important fact must not be forgotten. The fact that even if the robot is calibrated it will always be affected by some error factors (different sources) and that the real positions of the robot will be slightly (but still important) different from the ones displayed on the teach-pendant. For these reasons, working with the virtual models, pendant or applications is always spreferred due to the fact that the access to the robot or even the robot itself is not necessary, there is no need of interrupting a robot from work, and the values displayed in the virtual programming and simulation software will always be the exact ones. In Fig. 7 the virtual teach-pendant is presented.

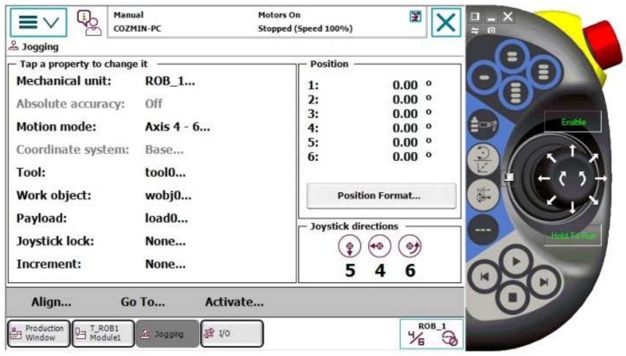


Fig. 7. Virtual teach-pendant from ABB RobotStudio.

All teach-pendant’s features plus many more are available also when using the graphical interface of the application. Currently, it is of interest to the subject of the paper that using the right commands we can position the robot in different configurations by setting joint angles and that points (“targets”) can be created in these positions with the possibility of visualizing their coordinates. It is further exemplified for the ABB IRB120 robot. Same steps needed to be applied similarly for the other two robot models.

First of all, the desired robot model must be loaded into the application. This is done by accessing the menu "File/Solution with station and Robot Controller" followed by the command "Create". When the application loads the necessary resources into the

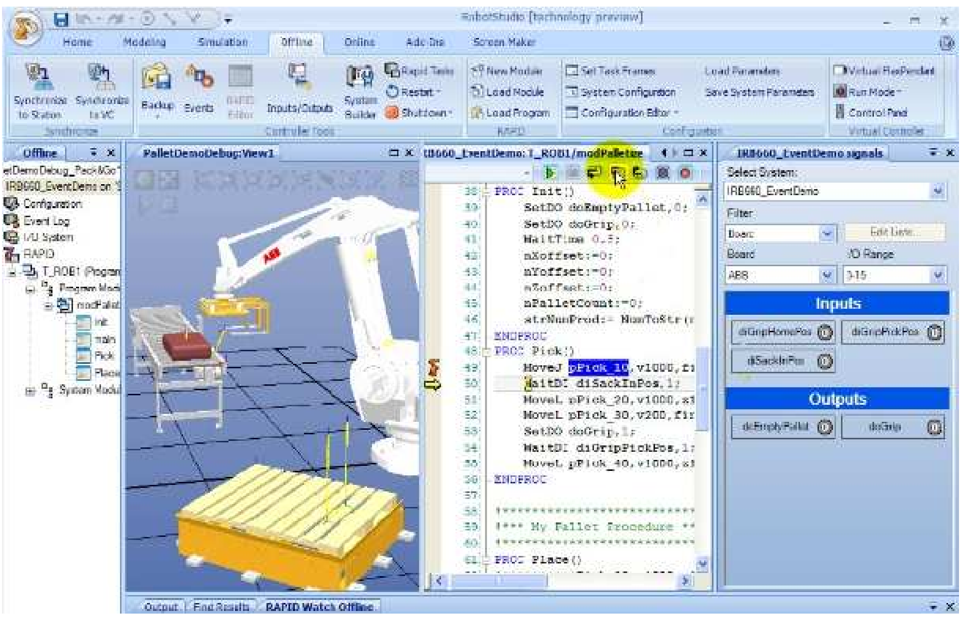


Fig. 6. ABB RobotStudio offline programming and simulation interface example [6].

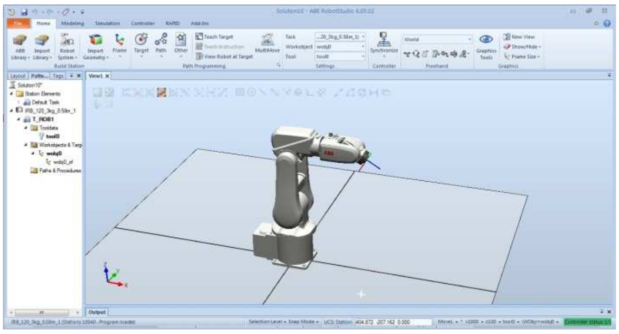


Fig. 8. IRB 120 Robot model loaded into an empty station.

simulation window, the selected robot will appear, centered in an empty cell ("station"). In addition to the simulation and programming windows and command toolbars, another very important element is the specifications tree (usually on the left side), which contains information about station components, robot equipment information, textual programs files, configurations, points and trajectories learned for the robot. The robot is currently loaded into an empty station and the specification tree will contain the information listed above when added or created (Fig. 8).

In order to be able to test the geometric model directly by comparing with results obtained from ABB RobotStudio, we must proceed in the same manner as presented in Chapter 2. In RobotStudio the robot is brought into the two configurations by changing the angles of the joints. Of course, the same joint values must be set in the parameter tables for the mathematical calculation and then compared the results. As in Chapter 2, for demonstration, the values for the same two positions of the robot will be compared: the "home" position and the "random" position.

Using the "Create joint target" command, the second position of the robot is created (the "random" position). At this step no points are created in those positions but the robot joint values are stored. In order to be able to measure the coordinates of the points where the robot arrives by applying those values to the joints, the command "Teach target" must be used. When using this command, the program creates a point exact in at that location where the robot is, point which can be seen in the simulation window (represented by an axis system) and can also be seen in the left specification tree. By accessing the properties of the point from specification tree its coordinates can be viewed or modified.

In the same manner as in previous illustrated steps and in Figs. 9, 10 and 11, the configurations for the three robots are stored in the program. By using the "Teach target" command for each configuration separately, the characteristic points for the respective configurations will be recorded, which will also appear in the specification tree and for which coordinates will be available. The coordinates of the points recorded in ABB RobotStudio for the same two configurations are further compared with the mathematically computed coordinates of the forward geometric model implemented in the Excel files (figures 12, 13 and 14).

Analyzing the results from ABB RobotStudio and comparing them with the results calculated for the forward geometric model we can see that for the same

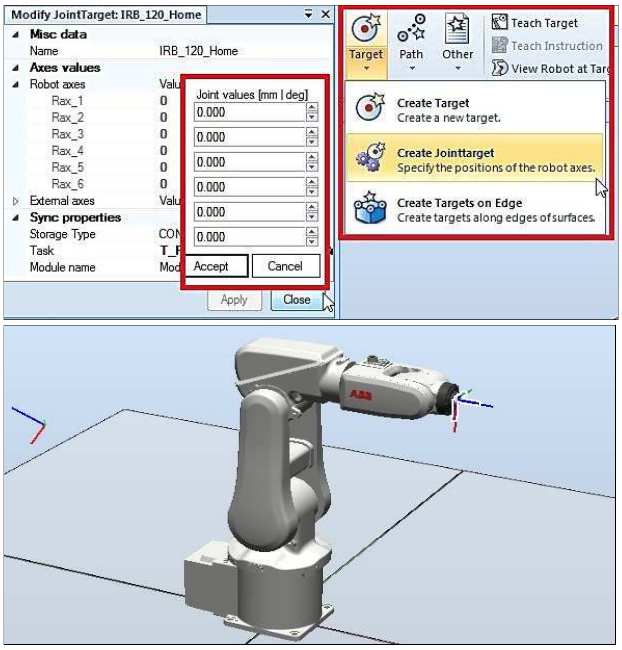


Fig. 9. Robot model at joint target for "home" position.

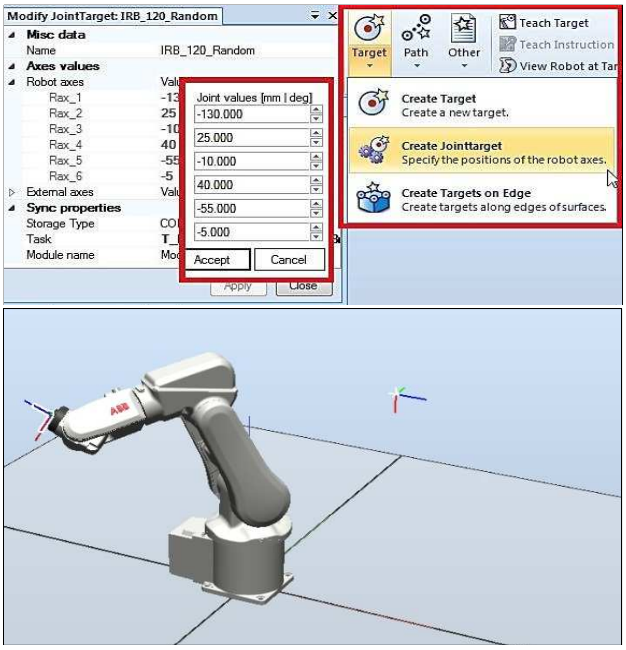


Fig. 10. Robot model at joint target for "random" position.

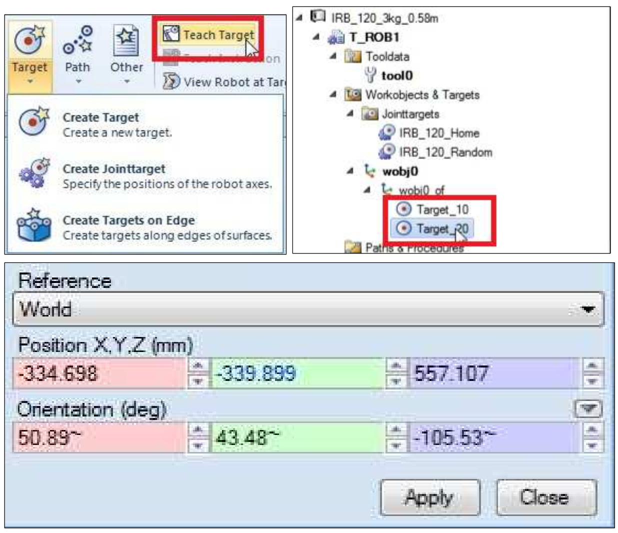


Fig. 11. Point coordinates for second pose ("random position").

robot configurations (the same joint angles) the coordinates of the characteristic point are identical. In other words, the geometric model is correct. If the results were different, it would have suggested mistakes in defining geometric model parameters or calculus errors. Since ABB RobotStudio is the official application for off-line programming and simulation of ABB robots and its virtual models are the ideal robot models we have the certainty that the values obtained with it are the right ones. If in the development of a forward geometric model after performing the necessary mathematical calculations are obtained same values as those obtained with ABB RobotStudio, we can be sure that the geometric model is a correct one.

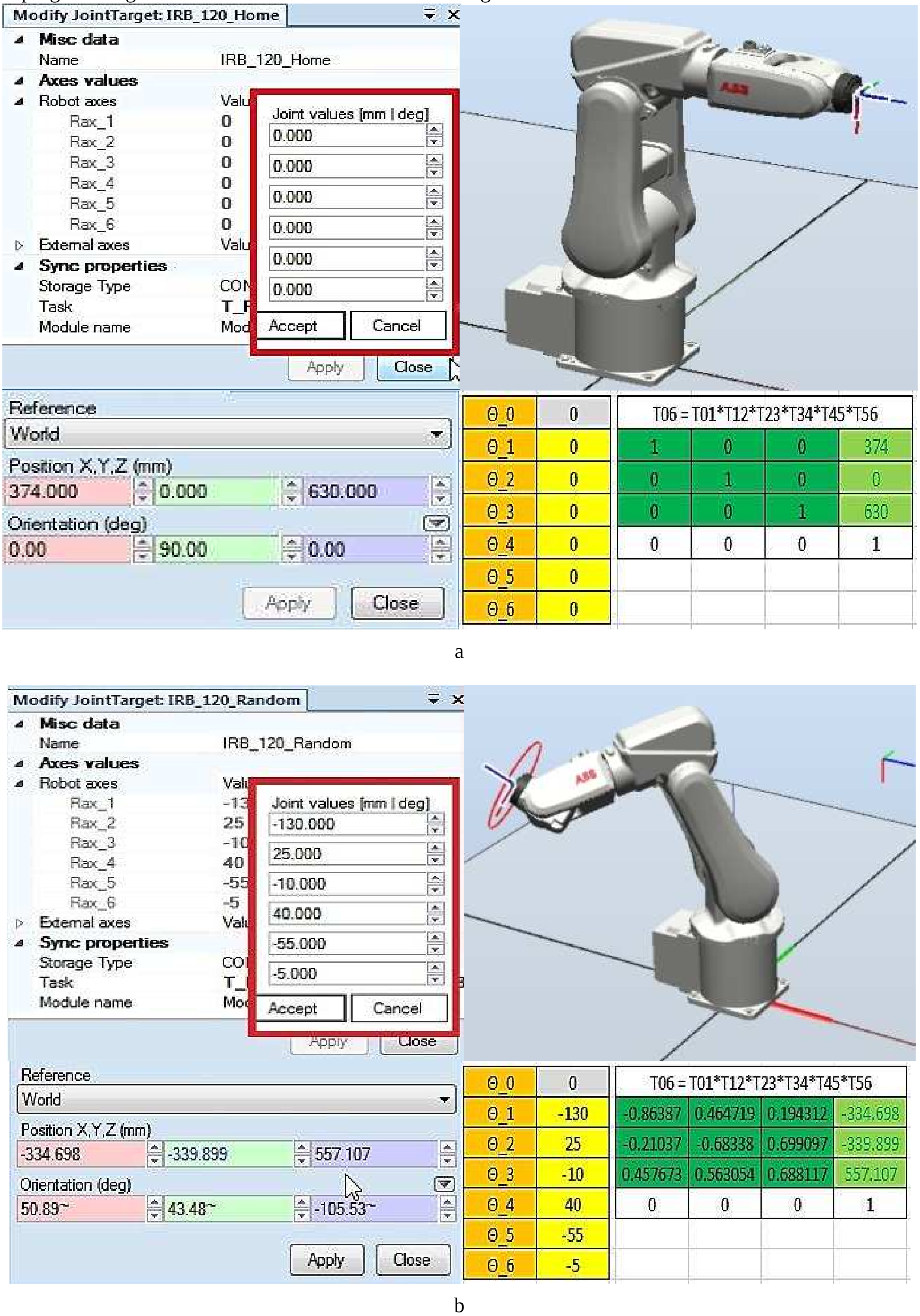
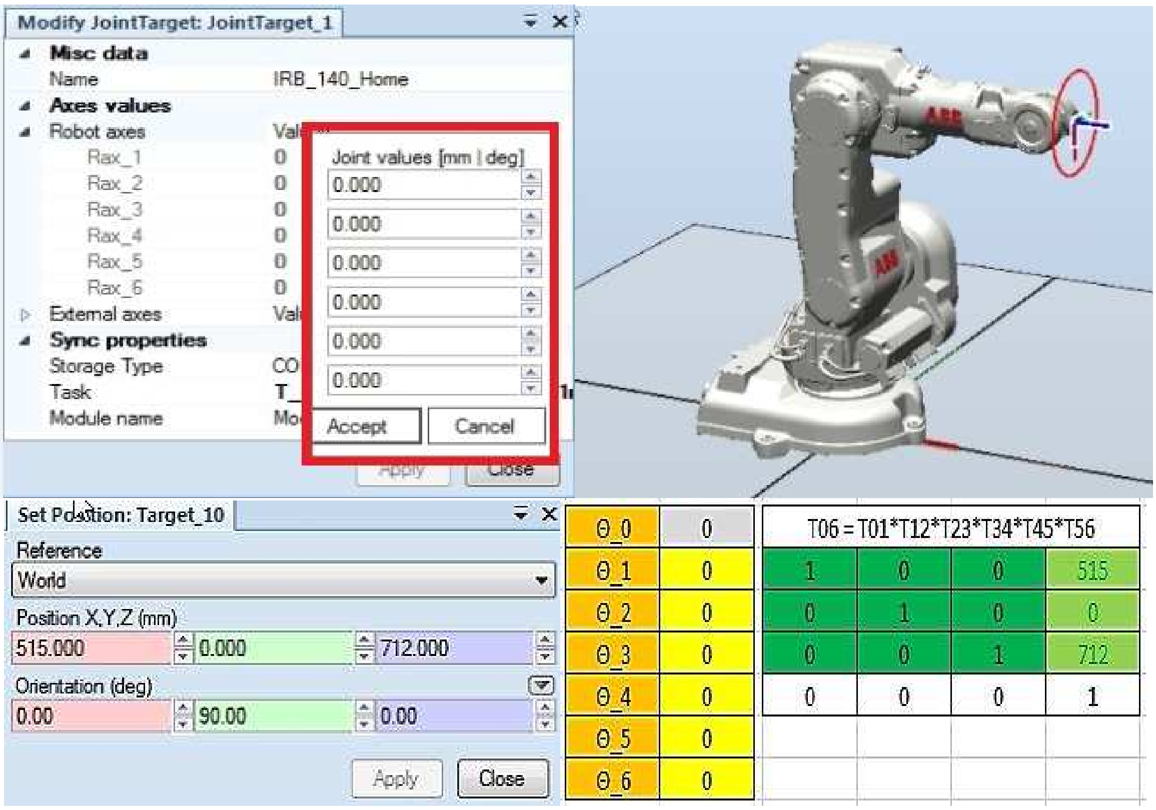
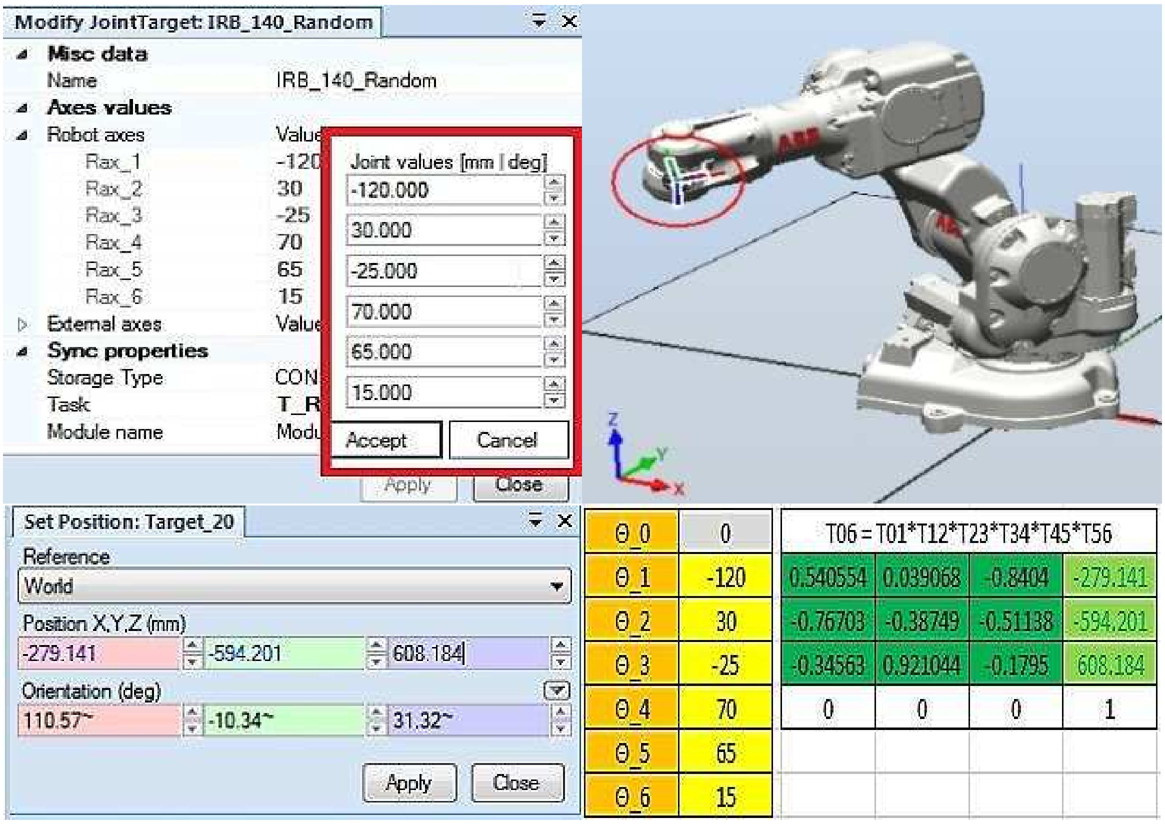


Fig. 12. ABB RobotStudio results and computed results for IRB 120: *a* – "home" position and *b* – "random" position.

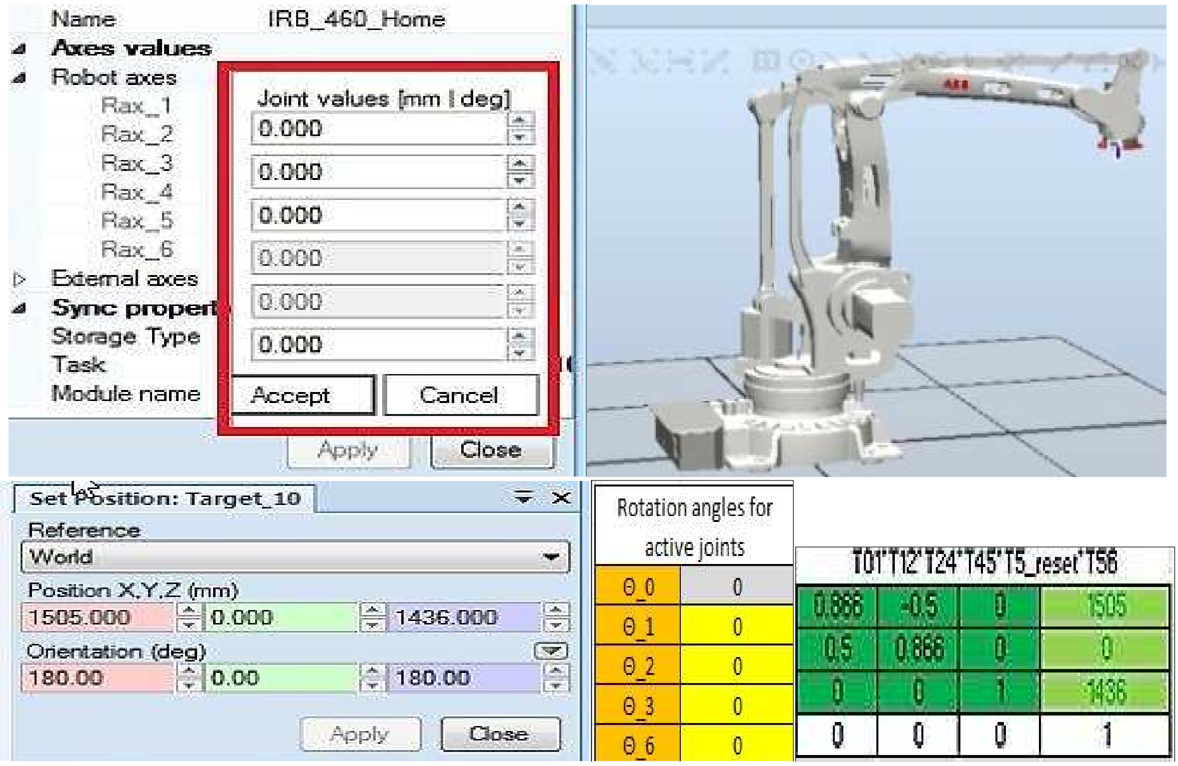


a

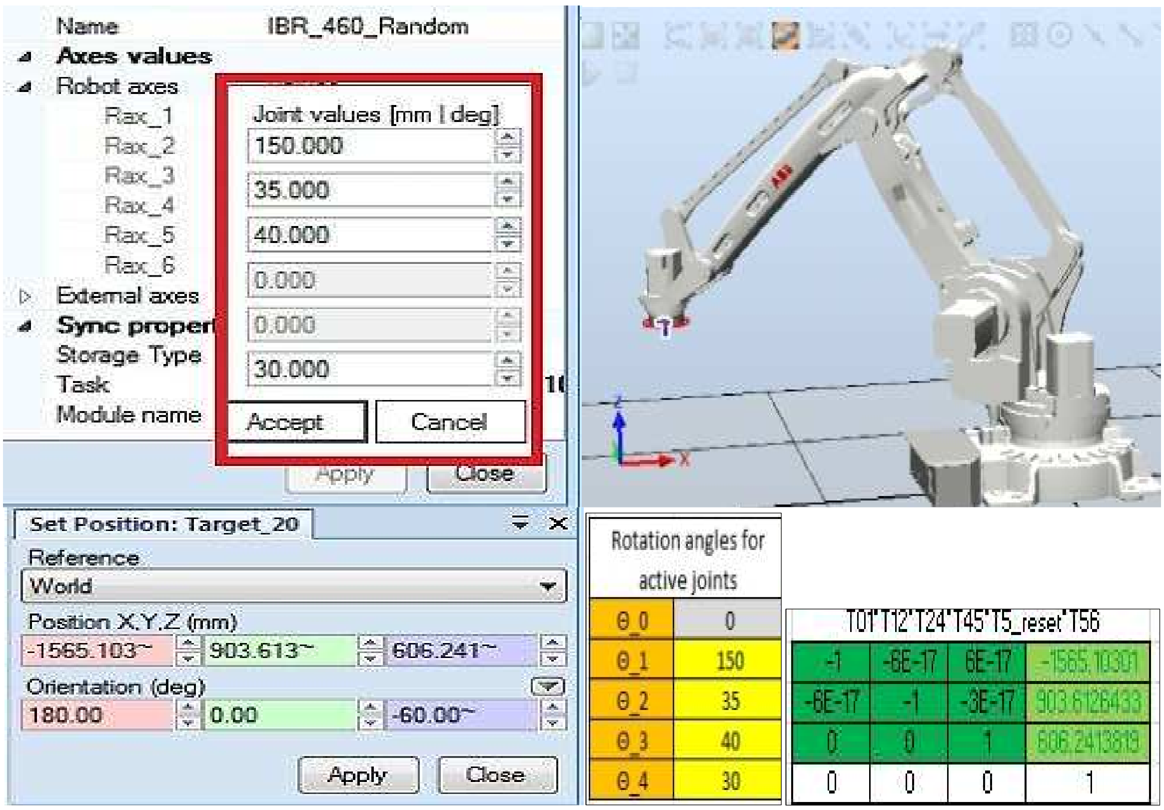


b

Fig. 13. ABB RobotStudio results and computed results for IRB 140: *a* – "home" position and *b* – "random" position.



a



b

Fig. 14. ABB RobotStudio results and computed results for IRB 460: *a* – "home" position and *b* – "random" position.

4. CONCLUSIONS

Improving performances of industrial robots is a constant need. One solution is improving the mathematical models of the robots by taking into account more than just some link dimensions parameters. Such models have been studied and presented in [7, 8, and 9] where forward geometric models were developed including real constructive and functional parameters, parameters for geometric or non-geometric errors (backlash, thermal etc.) proving that custom particular and improved forward geometric robot models can be used in developing some solutions for compensating these errors.

In paper [1], a new approach for forward geometric modeling of IR was presented. For the two open chain cinematic structures of the studied robots (ABB IRB 120 and IRB 140) the mathematical modeling results were compared with the results obtained from the classic Denavit–Hartenberg (DH) formalism proving the compatibility of the new approach with the D-H one. The correctness of the new approach was also confirmed by preliminary comparison and validation with the results from the robot virtual model using CATIA V5 CAD environment.

In paper [2] the new approach for forward geometric modeling of IR was also presented regarding the elaboration of the mathematical model for a closed cinematic structure IR (ABB IRB 460) usually used for palletizing operations. This second approach also involves modeling the passive joints and taking into account the elements of the closed kinematic chains. For the first time, a mathematical model in which not only the position of the active joints but also the positions of the passive joints are monitored was presented. This is of great importance if there is a need to develop a model that takes into account real functional constructive parameters. For example, the thermal deformations that may occur in the elements of the closed kinematic chain structure can seriously affect the positioning accuracy of the robot as well as the orientation of the axis 6 which for this type of robots must mandatory be in a vertical position to ensure the end flange parallelism with the ground. Applying the approach presented in papers [1, 2] the development of a complete mathematical model, where error parameters can easily be introduced as simple displacements in the parameter table (chapter 3, Table 2) becomes possible. This method is also more easily to be applied by using reference frames with same orientation on the robot's entire structure. This fact is also avoiding DH ambiguities about positioning of reference frames and provides a clear general method to be used by everyone. The mathematical results were computed within an Excel file and compared with the robot's virtual model. The results were identical proving the new method is correct.

Among the major benefits of applying the new approach we can avoid the ambiguities introduced by DH modeling with regard to the development of kinematic schemas and the assignment of reference systems. The fact that the coordinate systems now have the same orientation facilitates the elaboration of correct cinematic schemas in particular if it is desired to take into account of the real robot geometric parameters. In this way it is very easy to introduce parameters regarding to joint distances and displacement of the elements as simple translation parameters in the same directions *X*, *Y*, *Z*.

Thus, validation methods presented in this paper were used for final checking the forward geometric models developed by applying the new original approach presented in [1 and 2]. Supplementary to validation with other forward geometric models already existing in related literature presented in [1 and 2], once with chapter 3 of present paper a 2 way cross validation is completed by mean of:

1) a computer aided design software that is capable to load virtual models in STEP or IGS format (in this case CATIA);
2) the official programming and simulation software application for ABB robots – "ABB RobotStudio".

Every time the computed results of mathematical relations implemented in excel files were identical to those obtained from other models, results from the new approach modeling, CATIA software modeling and ABB RobotStudio software proving that the new forward geometric models are correct. In future studies in the rame of PhD thesis, these forward geometric models will be used and improved by adding error parameters (especially thermal error parameters) with the final goal of developing a software compensation model.

REFERENCES

[1] C. Cristoiu, A. Nicolescu, *New approach for forward kinematic modeling of industrial robots*, MIT Conference, Sinaia, 2017.

[2] C. Cristoiu, A. Nicolescu, *New approach for forward kinematic modeling of industrial robots with closed kinematic chain*, MIT Conference, Sinaia, 2017.

[3] ***ABB Product specification IRB120; Document ID: 3HAC035960-001; Copyright 2010-2017 ABB; www.abb.com/robotics.

[4] ***ABB Product specification IRB140; Document ID: 3HAC041346-001; Copyright 2010-2017 ABB; www.abb.com/robotics.

[5] ***ABB Product specification IRB460; Document ID: 3HAC039611-001; Copyright 2010-2017 ABB; www.abb.com/robotics.

[6] *** http://new.abb.com/products/robotics/RobotStudio/

[7] A. Nubiola, I.A. Bonev, *Absolute calibration of an ABB IRB 1600 robot using a laser tracker*, Robot Comp Integr Manuf 2013; Vol. 29, No. 1, pp. 236–245.

[8] A Cezara, E. Strajescu, *Research on structural and functional optimization of IR's numerically controlled axes in order to increase their performances*, PhD Thesis, University "Politehnica" of Bucharest, 2014.

[9] A.F. Nicolescu, F.M. Ilie, T.G. Alexandru, *Forward and inverse kinematics study of industrial robots taking into account constructive and functional parameters modeling*, Proceedings in Manufacturing Systems, Vol. 10, Issue 4, 2015, pp. 157−16.