DYNAMIC ANALYSIS AND VISUALIZATION OF SPATIAL MANIPULATORS WITH CLOSED STRUCTURE

Vjekoslav Damic^a, Maida Cohodar^b

^a*University of Dubrovnik, Cira Carica 4, Dubrovnik 20000, Croatia*

^b*Faculty of Mechanical Engineering, University of Sarajevo, Vilsonovo setaliste 9, Sarajevo 71000, Bosnia and Herzegovina*

Abstract

Visualisation becomes powerful tool in dynamic analysis of engineering application. Attention of the paper is focused on technique of visualization of closed-chain manipulators using BondSimVisual. The basic idea of the paper is to develop the models of manipulators from two points of view: the dynamic and geometric. The dynamic model is developed using a suitable bond graphs modelling and simulation environment (BondSim). To develop 3D model in a virtual scene another program the BondSimVisual is used. It is based on VTK visualization pipeline technology. These two models interact by two-way IPC (inter process communication) based on the named pipes. Thus, the two models appear as a single complex model running on the same or separate computers connected by a LAN. During the simulation, the models (the dynamic and visual) exchange the necessary information. The proposed approach is illustrated on two examples.

Keywords: visualization; bond graphs modelling; dynamics; VTK

This Publication has to be referred as: Damic, V[jekoslav] & Cohodar, M[aida] (2016). Dynamic Analysis and Visualization of Spatial Manipulators with Closed Structure, Proceedings of the 26th DAAAM International Symposium, pp.0109-0117, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-07-5, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/26th.daaam.proceedings.016

1. Introduction

Object oriented approach to system dynamic modelling and visualization are announced as the future trends in analysing behaviour of the complex systems [1,3-7,13,15]. With help of system visualization detection of possible collisions between parts of complex systems and recognition of singularity postures in order to avoid them become much simpler. Visualisation contributes to better understanding of a system behaviour and its functionality. These are the reasons why development of visualization toolkits is subject of many investigations today. Attention of the paper is focused on technique of visualization of closed-chain multibody systems using BondSimVisual (developed by the first author of the paper). It is based on Visualization Toolkit C++ library [12]. VTK is recognised as flexible and powerful toolkit in [2], where it is proposed framework for use in MATLAB programming environment.

The virtual model is used in synergy with dynamic model in analysis of behaviour of multibody systems in this paper. The dynamic model is developed by bond graphs using BondSim program [8]. Thus, two models of the same multibody system, created using two different software packages, can run on the same or different computers during the simulation. Communication between them is based on named pipe technology [9,10]. The paper is organized as follows: Concept of visualization by BondSimVisual is generally explained in section 2. It is applied on example of a four bar mechanism in section 3, and real robot manipulator in section 4. The paper finishes with concluding remarks and announcement of the future work.

2. Concept of visualization

2.1. Development of visual model

BondSimVisual is based on Visualization Toolkit (VTK) C++ library. It is platform independent, very powerful and freely available toolkit [12]. The virtual model of a multibody system is defined by a script which can be edited using any pure text editor such as Notepad. In script file, we firstly need to define the kinematical structure of the mechanism. The global (World) 3D coordinate system is assumed as XYZ system with X-axis horizontally to the right, Y-axis vertical up, and Z-axis out of the computer screen. The Robot command defines first position and orientation of the base coordinate system with respect to the global, and then defines joint per joint going from the base to the robot tip. It is also possible to define the closed loops of the links by command Extend. Each joint is defined with respect to the previous one basically in the form of homogenous transformation of the joint frames. The Robot command can be used for any mechanism not only the robots, but the robots are most demanding.

To define parts of the multibody system two approaches are used:

- Using 3D CAD models of multibody parts in form of stl (stereo lithography) files, and
- Using basic shapes: cubes, cylinders, poly-cylinders, spheres, poly-prisms, modules etc.

In the first approach, 3D CAD models of each part of a multibody system are developed and exported in form of stl file using a 3D CAD software such as CATIA, Pro Engineer, Solid Works, etc. Many robot manufacturers already offer 3D models of robot parts in different formats including stl format and allow their free use. BondSimVisual supports the stl format using the corresponding VTK's reader.

In the second approach the parts are built directly by the script commands of the primitive parts such as cubes, cylinders, spheres, poly-prisms, and modules. It is necessary in this case to have detailed information of the system geometry. The parts are transformed by rotating and translating them and may be assembled into the sets, and used to define the multibody system.

Finally, the rendering of the system parts is defined [9,10]. Here the colors of the parts can be defined. The script for a problem is imported into BondSimVisual program as text files.

2.2. Development of the dynamic model and communication between two models

The dynamic model is developed by BondSim program, which is based on bond graphs. It supports the inter-process communications (IPC) based on the named pipes. In this paper the closed-chain multibody systems are developed as interacting geometric (virtual) and dynamic models.

The visualization really starts by opening a project script on the visual side (BondSimVisual). The program reads the script, translates it and generates the objects in the memory, which constitute the visualization pipeline for the problem following the VTK approach. It is used to render the system on the computer screen in its initial configuration. But later, it is used to redraw the virtual scene as client (BondSim) asks. The system creates also a server side of inter-process communication pipe and a listening thread, asks the client to connect and start using the visual model, i.e. conduct the visualization.

On the dynamical side, i.e. in BondSim, the program is first loaded and then the corresponding model opened and built. The client side pipe is created, and communication with the visual side established. The simulation now can start. During the simulation two models can run and interact on same or different computers connected by the local net. When the simulation begins the dynamic model regularly, e.g. every 20 ms, sends data to the visual model and

effectively drives it over the virtual scene. Simultaneously the visual model returns back the information defined in the script. These are usually the coordinates of points in the geometrical objects as they move over the 3D scene. These information can be used in dynamical model to simplify solution of some problems, e.g. a clash between the objects in the system.

For both developed models in the paper – four bar mechanism and industrial robot ABB IRB 4400, the dynamic model sends position information (joint angles) and receives information about coordinates of predefined point. In these examples they are imported into the dynamic side to compare them with ones evaluated by the dynamic model in order to check validity of virtual models and established communications between two models.

3. Development of visual model of a four bar mechanism

A four bar mechanism, shown in Fig. 1, consists of three bodies: the crank shaft, connecting rod and follower. They are connected to each other and to the base by the revolute joints.

Although bodies of four bar mechanism form the closed structure having one degree of freedom, the virtual model of the mechanism is created as an open chain. The closed structure of this mechanism can be opened in different ways. Note that in any case the dynamical model is closed and that it supplies the visual model with values of all joint variables. The visual model only follows the commands received from the dynamical side. In this paper two forms of open chain structure are visually modeled and analyzed. According to the first, the mechanism is opened at point D and considered as a serial chain of the bodies (model M1). In the second case the mechanism is opened at the point C, and it is composed of two parallel structures (model M2).

Fig. 1. (a) Four bar mechanism - scheme; (b) Dynamic model of Four bar mechanism; (c) Driven torque

Dynamic model of mechanism is the same for both cases, of course. It is developed using bond graphs. It is taken from [11] and just slightly modified. IPC component is added to model (a ring at the bottom-right), Fig. 1b.

The crank shaft is driven by torque:

$$T(t) = \begin{cases} 10 \sin(3\pi t) & 0 \leq t \leq 5/18 \\ 465.8838298 \cdot e^{-16.324194 \cdot t} & t > 5/18. \end{cases} \quad (1)$$

The torque developed according to Eq. (1) is shown in Fig.1c. The dynamic model calculates corresponding values of the rotation angle of the crank shaft. Values of rotations of the others parts are evaluated on the dynamic side as well. These signals are also connected to IPC component and sent to the virtual model. For virtual model M1 the values of the joint rotation angles θ_1 , θ_2 and θ_3 (Fig.1a) are sent, and for the model M2 the values of θ_1 , θ_2 and θ_4 angles (Fig.1a). When the virtual model receives the data, it redraws the virtual scene. There are no any other calculations on the virtual side, only the reposition of the objects on the screen. Material and geometric parameters of four bar mechanism are given in Table 1. Simulation was performed for time of 1 s, and output interval of $1e-4$ s with absolute and relative error tolerance $1e-8$ s. The delay, which defines when the data are sent to the virtual side, is set to 5 ms, i.e. every 5 ms the scene is refreshed.

Parameter	Crank shaft	Connecting rod	Follower
Length L [m]	0.2	0.9	0.5196152
Cross section area A [m ²]	1.257e-3	1.96e-3	1.257e-3
Mass moment of inertia I [kgm ²]	2.27e-3	1.669e-1	5.88223e-3

Table 1. Material and geometric parameters of Four bar mechanism

3.1. Virtual model M1

The script file for visualization of four bar mechanism, model M1 is presented in Appendix A. The virtual model is developed using the base coordinate frame coincident with the world frame. Using the instruction Robot the virtual model named Four_bar_mechanism_M1 is defined. The next three Joint instructions define that the mechanism has three successive joints. Coordinate frame is attached to the each joint. Coordinate frame, denoted by index 0 is fixed, and others are denoted by indexes 1, 2, and 3. The arguments of instruction Joint i provide transformation coordinates from (i-1) to i th frame. The attached bodies – Crank Shaft, Connecting rod and Follower are defined and can move as the joints permit.

The second and the third joint are shifted by 200 and 900 mm along the corresponding x-axis of the local frame, providing zero values of joint angles θ_1 , θ_2 and θ_3 , as shown in Fig. 2a. The initial position of the mechanism is different and is defined by $\theta_{1in}=0^\circ$, $\theta_{2in}=30^\circ$ and $\theta_{3in}=-150^\circ$, as shown in Fig. 2c. When simulation starts dynamic model sends these initial values of joint angles to the visual side, and after the virtual scene is redrawn the mechanism starts the motion from that position (Fig.2c).

Fig. 2. (a) Start position of M1; (b) Start position of M2; (c) Initial position

Simulation results are presented in Fig. 3. The values of X and Y coordinates of mid-point M of the connected rod, evaluated on the dynamic side (denoted by index M) are compared with coordinates received from virtual model (denoted by index Mv). Fig.3a shows they are visually nearly identical.

As consequence of three rotations by angles θ_1 , θ_2 and θ_3 , which are calculated on dynamic side, we obtain that free end of the last link of the open chain (the follower) practically does not move. It behaves as it is attached to the base by the revolute joint. Values of this point coordinates (in millimetres) are imported from the virtual model and presented on display, Figs. 3b and 3c.

Fig. 3. (a) Comparison of coordinates of mid-point M calculated on the dynamic side with ones received from the virtual model; (b) X coordinate of point D imported from virtual model; (c) Y coordinate of point D imported from virtual model

3.2. Virtual model M2

The script file for visualization of M2 is presented in Appendix A. In the case of model M2, the four bar mechanism is opened at point C and the model has two parallel structures connected to the base. The left side of the model consists of two revolute joints A and B . The right part is connected by the joint D to the base. The base is

attached the ground by immobile joint 0. To create the branched structure the *Extend* instruction is used. It simply defines the joint where the right branch starts. The zero position of the four bar mechanism (with zero values of rotation angles θ_1 , θ_2 and θ_4), is depicted in Fig.2b. As in the previous case, when simulation starts, the dynamic model sends real initial values of these angles ($\theta_{1in}=0$, $\theta_{2in}=30^\circ$ and $\theta_{4in}=60^\circ$). Simulation results are shown in Fig.4.

Fig. 4. Comparison of coordinates of point C: (a) calculated on the dynamic side with ones received from the virtual model from left; (b) calculated on the dynamic side with ones received from the virtual model from right branch

The coordinates of point C imported from the virtual side are compared with values of coordinates calculated by the dynamic model. The coordinates of point C are shown separately for the left branch (denoted by index *Cvl*, Fig.4a), and for the right branch (denoted by index *Cvr*, Fig.4b). The simulation shows a good agreement.

4. Models of industrial robot ABB IRB4400

The proposed approach is also applied to industrial robot ABB IRB 4400. It consists of an anthropomorphic arm and a spherical wrist and has 6 rotation joints. A parallelogram links are applied between the shoulder and the elbow joints. It assures that the rotation angle of the third joint is $(\theta_3-\theta_2)$, where θ_i is rotation angle of i^{th} rotation axis. The frames are attached to each link (Fig. 5) according to Denavit Hartenberg (DH) notation, and DH parameters are specified in Table 2.

Fig. 5. ABB IRB4400 with attached frames

Link i	a_i	α_i	d_i	θ_i
1	a_1	$-\pi/2$	L_1	θ_1
2	L_2	0	0	$\theta_2 - \pi/2$
3	L_3	$-\pi/2$	0	$\theta_3 - \theta_2$
4	0	$\pi/2$	L_4	θ_4
5	0	$-\pi/2$	0	θ_5
6	0	0	L_6	$\theta_6 + \pi$

Table 2. DH parameters for IRB4400

The homogeneous transformation matrices for the joints are:

$$\begin{aligned}
 \mathbf{A}_1^0 &= \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & a_1 \cos(\theta_1) \\ \sin(\theta_1) & 0 & \cos(\theta_1) & a_1 \sin(\theta_1) \\ 0 & -1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{A}_2^1 = \begin{bmatrix} \sin(\theta_2) & \cos(\theta_2) & 0 & L_2 \sin(\theta_2) \\ -\cos(\theta_2) & \sin(\theta_2) & 0 & -L_2 \cos(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 \mathbf{A}_3^2 &= \begin{bmatrix} \cos(\theta_3 - \theta_2) & 0 & -\sin(\theta_3 - \theta_2) & L_3 \cos(\theta_3 - \theta_2) \\ \sin(\theta_3 - \theta_2) & 0 & \cos(\theta_3 - \theta_2) & L_3 \sin(\theta_3 - \theta_2) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{A}_4^3 = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 0 \\ 0 & 1 & 0 & L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 \mathbf{A}_5^4 &= \begin{bmatrix} \cos(\theta_5) & 0 & -\sin(\theta_5) & 0 \\ \sin(\theta_5) & 0 & \cos(\theta_5) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{A}_6^5 = \begin{bmatrix} -\cos(\theta_6) & \sin(\theta_6) & 0 & 0 \\ -\sin(\theta_6) & -\cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & L_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
 \end{aligned} \tag{2}$$

Computation of the direct kinematic function yields:

$$\mathbf{A}_6^0 = \mathbf{A}_1^0 \cdot \mathbf{A}_2^1 \cdot \mathbf{A}_3^2 \cdot \mathbf{A}_4^3 \cdot \mathbf{A}_5^4 \cdot \mathbf{A}_6^5. \tag{3}$$

Values of parameters are: $a_1=0.2$ m, $L_1=0.68$ m, $L_2=0.89$ m, $L_3=0.15$ m, $L_4=0.88$ m and $L_6=0.14$ m [15]. Each joint is driven by the angular velocities corresponding to the following joint angles (in radians):

$$\theta_i = -k_i \cdot \pi \cdot \sin\left(\frac{\pi t}{10}\right), \quad k_1 = \frac{5}{6}, k_2 = \frac{1}{6}, k_3 = \frac{1}{18}, k_4 = k_6 = 1, k_5 = \frac{1}{2}. \tag{4}$$

Two models are developed again – the dynamic and virtual (Fig. 6). Owing to object oriented approach of BondSim, the dynamic model of manipulator IRB4400 (Fig. 6a) is developed using dynamic model of IRB1600 robot [10]. The model was modified taking into account the existence of the closed structure. Geometric and mass parameters were updated in the model as well. The manipulator parts have the complex shapes, and hence to develop virtual model the first proposed modeling approach was used. 3D CAD models of links in form of *stl* files are taken from [15]. A script was generated which defines the configuration of the robot, and the position and orientation of each links. Script file is given in Appendix B. The parts are defined using the same names as the CAD files.

Fig. 6. ABB IRB4400: (a) Dynamic model; (b) Visual model

Real manipulator IRB4400 consists of eight joints. Six of them are the basic, driven by the drives, which generates the joint angles defined by Eq.(4). Other two joints create closed structure assuring a better rigidity and accuracy of the manipulator. However, the virtual model is developed as open structure with seven joints. The last one was opened. An instruction *Extend* is used to define that the parallel loop (dark gray in Fig. 6b) starts at the seventh joint (Joint 6 in the script). The dynamic model via *IPC* component sends values for all seven joints. Manipulator also has pair of moving parts, but they are not taken in consideration to simplify the description of the proposed visualization procedure.

Simulation time was 25 s. Manipulator mass is 1040 kg [15]. Unfortunately, we did not have the dynamic parameters of the robot. The mass of each link was obtained by dividing the total robot mass proportionally to the link volumes. The link volumes are on the other hand determined by loading 3D CAD models of the links into CATIA program. The other parameters are generated in this manner as well. To verify the virtual model and established two way communication between dynamic and virtual model, the values of coordinate of the tool center point TCP (denoted by index *V*) are imported to the dynamic side, and compared with the values generated by the dynamic model (denoted by index *D*), as well with the exact analytical solution obtained by Eq.(3) (the first three elements of the fourth column of the homogeneous transformation matrix). The results of the comparison are shown in Fig. 7, from which we can see a good agreement between coordinates obtained on all three methods. From an enlarged figure on Fig. 7c it can be seen that information obtained from virtual model are shifted along the time axis. This is time needed for the data transfer between the models.

Fig. 7. TCP point: (a) X-coordinate; (b) Y-coordinate; (c) Z-coordinate

5. Conclusion

The attention in this work was devoted to the visualization of the robot with the closed links in form of the parallelogram structures. The manufactures often use this to improve higher payload capabilities of the robot, and good accuracy. It was shown that such structures can be efficiently dynamically simulated using Bond Graphs and generated their virtual 3D models. Proposed approach is applied to example of a four bar mechanism as simple planar case and the real industrial robot ABB IRB 4400.

The visual models were developed using software package BondSim Visual, based on VTK C++ library. Their dynamical model was created by BondSim program. During the simulations the communication between two models were achieved using named pipe technology. This provides analysis of the system behaviour from the both side – in dynamical space and in 3D virtual scene. The visualization generally improves the analysis of the behaviour of robots and other multibody systems. The *IPC* technology based on named pipes and 3D virtual modelling opens now possibilities to modelling of mechatronics system on a larger scale. It will be one of main directions for future investigation.

6. Acknowledgements

This paper is realized in the framework of a project supported by the Federal ministry of education of Bosnia and Herzegovina.

7. Appendix A. Script files for visualization of Four Bar Mechanism

Model M1	Model M2
!----- Four_bar_mechanism -----	!----- Four_bar_mechanism_M2 -----
Robot Four_bar_mechanism	Robot Four_bar_mechanism_M2
Joint 1 revolute Z	Joint 0 prismatic x (shift y-20)
Joint 2 (shift x 200) revolute Z	Joint 1 (shift y 20) revolute Z
Joint 3 (shift x 900) revolute Z;	Joint 2 (shift x 200) revolute Z


```

cube T0000170 50.000000 40.000000 50.000000 ;
cube T0000171 50.000000 40.000000 50.000000 ;
cube T0000172 869.615200 40.000000 50.000000 ;
cylinder T0000173 diameter 10.000000 resolution 16 length
34.740000;
cylinder T0000174 diameter 10.000000 resolution 16 length
34.740000;
set Base add T0000170 (shift x-25 y-20 z -50)
    T0000171 (shift x 694.6152 y-20 z-50)
    T0000172 (shift x -75 y-60 z-50)
    T0000173
    T0000174 (shift x 719.6152 );
Set Four_bar_mechanism add Base;
Render Four_bar_mechanism
    color 0.91 0 0;
cube T0000175 220 36.286 34.64 ;
cylinder T0000176 diameter 10.000000 resolution 16 length
48.340000;
Set CrankShaft add T0000175 ( shift x -10 y -18.143)
    T0000176 (shift x 200 z 34.64);
Set Four_bar_mechanism#1 add CrankShaft ;
Render Four_bar_mechanism#1
    color 0.0 0.6 0.6 ;
cube T0000177 920 45.224 43.34;
Set ConnectingRod add T0000177 (shift x -10 y-22.612 z
34.62 );
Set Four_bar_mechanism#2 add ConnectingRod ;
Render Four_bar_mechanism#2
    color 0.0 0.3 0.3 ;
cube T0000178 539.6152 36.286 34.64;
cylinder T0000179 diameter 10.000000 resolution 16 length
48.340000;
Set Follower add T0000178(shift x-10 y -18.143 )
    T0000179 (shift z 34.64);
Set Four_bar_mechanism#3 add Follower ;
Render Four_bar_mechanism#3
    color 0.8 0.8 0.8 ;
Probe Point1 Four_bar_mechanism#2(shift x 450) refer
Four_bar_mechanism;
!----- End -----

```

```

Extend Joint 0 Joint 3 (shift x 719.6152 y 20) revolute Z;
cube T0000172 869.615200 40.000000 50.000000 ;
set Base0 add T0000172 (shift x -75 y-60 z-50);
Set Four_bar_mechanism_M2 add Base0;
Render Four_bar_mechanism_M2
    color 0.91 0 0;

cube T0000170 50.000000 40.000000 50.000000 ;
cube T0000171 50.000000 40.000000 50.000000 ;

cylinder T0000173 diameter 10.000000 resolution 16 length
34.740000;
cylinder T0000174 diameter 10.000000 resolution 16 length
34.740000;
set Base add T0000170 (shift x-25 z -50)
    T0000171 (shift x 694.6152 z-50)
    T0000173 (shift y 20)
    T0000174 (shift x 719.6152 y 20);
Set Four_bar_mechanism_M2#0 add Base;
Render Four_bar_mechanism_M2#0
    color 0.91 0 0;

cube T0000175 220 36.286 34.64 ;
cylinder T0000176 diameter 10.000000 resolution 16 length
48.340000;
Set CrankShaft add T0000175 ( shift x -10 y -18.143)
    T0000176 (shift x 200 z 34.64);
Set Four_bar_mechanism_M2#1 add CrankShaft ;
Render Four_bar_mechanism_M2#1
    color 0.0 0.6 0.6 ;

cube T0000177 920 45.224 43.34;
Set ConnectingRod add T0000177 (shift x -10 y-22.612 z 34.62 );
Set Four_bar_mechanism_M2#2 add ConnectingRod ;
Render Four_bar_mechanism_M2#2
    color 0.0 0.3 0.3 ;
cube T0000178 539.6152 36.286 34.64;
cylinder T0000179 diameter 10.000000 resolution 16 length
48.340000;
Set Follower add T0000178(shift x-10 y -18.143 )
    T0000179 (shift x 519.6152 z 34.64);
Set Four_bar_mechanism_M2#3 add Follower ;
Render Four_bar_mechanism_M2#3
    color 0.415 0.517 0.572 ;

Probe Point1 Four_bar_mechanism_M2#2(shift x 900) refer
Four_bar_mechanism_M2;
!Probe Point1 Four_bar_mechanism_M2#3(shift x 519.6152) refer
Four_bar_mechanism_M2;
!----- End -----

```

8. Appendix B. Script files for visualization of ABB IRB4400

```

!----- ABB IRB4400_60 -----
Robot ABB_IRB4400_60 (euler -90.0 -90.0 90.0)
    Joint 0 revolute Z
    Joint 1 (shift x 200 z 680 ) revolute y
    Joint 2 (shift z 890) revolute y
    Joint 3 ( shift x 880 z 150 ) revolute x
    Joint 4 revolute y
    Joint 5 revolute x
    Extend Joint 0 Joint 6 (shift x -100 z 680) revolute y;
Part IRB4400_60_m2000_rev1_01-1_Body0;
Part IRB4400_60_m2000_rev1_01-2_Body1;
Part IRB4400_60_m2000_rev1_01-4_Body1_1;
Part IRB4400_60_m2000_rev1_01-3_Body2;
Part IRB4400_60_m2000_rev1_01-8_Body3;
Part IRB4400_60_m2000_rev1_01-10_Body4;
Part IRB4400_60_m2000_rev1_01-12_Body5;
Part IRB4400_60_m2000_rev1_01-13_Body6;
Part IRB4400_60_m2000_rev1_01-7_Body2_3;
Set ABB_IRB4400_60 add IRB4400_60_m2000_rev1_01-1_Body0;
Render ABB_IRB4400_60
    color 0.89 0.423 0.039;
Set ABB_IRB4400_60#0 add IRB4400_60_m2000_rev1_01-2_Body1
    IRB4400_60_m2000_rev1_01-4_Body1_1;
Render ABB_IRB4400_60#0

```

```

color 0.89 0.423 0.039 ;
Set ABB_IRB4400_60#1 add IRB4400_60_m2000_rev1_01-3_Body2 (shift x -200 z -680 );
Render ABB_IRB4400_60#1
color 0.89 0.423 0.039 ;
Set ABB_IRB4400_60#2 add IRB4400_60_m2000_rev1_01-8_Body3 (shift x -200 z -1570 );
Render ABB_IRB4400_60#2
color 0.89 0.423 0.039 ;
Set ABB_IRB4400_60#3 add IRB4400_60_m2000_rev1_01-10_Body4 (shift x -1080 z -1720 );
Render ABB_IRB4400_60#3
color 0.89 0.423 0.039 ;
Set ABB_IRB4400_60#4 add IRB4400_60_m2000_rev1_01-12_Body5 (shift x -1080 z -1720 );
Render ABB_IRB4400_60#4
color 0.89 0.423 0.039 ;
Set ABB_IRB4400_60#5 add IRB4400_60_m2000_rev1_01-13_Body6 (shift x -1080 z -1720 );
Render ABB_IRB4400_60#5
color 0.1 0.1 0.1 ;
Set ABB_IRB4400_60#6 add IRB4400_60_m2000_rev1_01-7_Body2_3 (shift x 100 z -680);
Render ABB_IRB4400_60#6
color 0.1 0.1 0.1 ;
Probe Point1 ABB_IRB4400_60#5(shift x 140) refer ABB_IRB4400_60;
!----- End -----

```

9. References

- [1] B. Zhu, A. Song, X. Xu, S. Li, Research on 3D Virtual Environment Modeling Technology for space tele-robot, *Procedia Engineering*, 99, (2015), 1171-1178.
- [2] E. Birngruber, R. Donner, G. Langsmat, VTK – 3D visualization for MATLAB, *Insight Journal* [<http://hdl.handle.net/10380/3076>] Distributed under Creative Commons Attribution License.
- [3] H. Elmqvist, S.E. Mattsson, C. Chapuis, Redundancies in Multibody Systems and Automatic Coupling of Catia and Modelica, *Proceeding 7th Modelica Conference*, pp. 551:560, Sep. 20-22, 2007, Como, Italy.
- [4] L. Zlajpah, Robot Simulation for Control Design, *Robot Manipulators Trends and Development*, Agustin Jimenez and Basil M Al Hadithi (Ed.), ISBN: 978-953-307-073-5, InTech, DOI: 10.5772/9206. 2010, Available from: <http://www.intechopen.com/books/robot-manipulators-trends-and-development/robot-simulation-for-control-design>.
- [5] M. Dupac, An object-oriented approach for mechanical components design and visualization, *Engineering and Computers*
- [6] N. Prabhu, M. Anand, S. Raja, Development of a Cad Based Platform for Scorbot_ER Vu Industrial Robot Manipulator, *Procedia Engineering* 38 (2012), 3992-3997.
- [7] N. Smith, C. Egert, E. Cuddihy, D. Walters, Implementing Virtual Robots in Java3D Using a Sudsumption Architecture, *Proceedings from the Association for the Advancement of Computing in Education*, 2006.
- [8] V. Damic, J. Montgomery, *Mechatronics by Bond Graphs: An Object-Oriented Approach to Modelling*. Berlin-Heidelberg: Springer-Verlag, 2003.
- [9] V. Damic, M. Cohodar, D. Damic, Multibody systems dynamical modeling and visualization based on IPC technique, *The proceedings of the 2014 Int. Con. on Bond Graph Modeling and Simulation – ICBGM'2014, Simulation Series*, The Society of Modeling & Simulation, Vol.46, No.8, pp. 773:778, July 6-10, 2014, Monterey, USA.
- [10] V. Damic, M. Cohodar, Dynamic analysis and 3D visualization of multibody systems, *IMAACA2015*, Sep.21-23, 2015, Bergeggi, Italy
- [11] V. Damic, M. Cohodar, M. Kulenović, Physical Modeling and Simulation of Mechatronics Systems by Acausal Bond Graphs, pp. 247-248, *Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium*, Volume 22, No. 1, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, 2011.
- [12] VTK User's Guide, Install, Use and Extend The Visualization Toolkit, Kitware, Inc., 11th ed., 2010.
- [13] W.J.Schroeder, K.M.Martin, W.E.Lorensen, The design and implementation of an object-oriented toolkit for 3D graphics and visualization, *Proceeding VIS, 7th Conference on visualization*, 1996, IEEE Visualization '96, 93-100.
- [14] S. Zodey, S.K.Pradhan, Matlab toolbox for kinematic analysis and simulation of dexterous robotic grippers, *procedia Engineering*, Vol.97, (2014).1886-1895.
- [15] <http://new.abb.com/products/robotics/industrial-robots/irb-1600/irb-1600-cad>