

Dynamic Simulation of a KUKA KR5 Industrial Robot using MATLAB SimMechanics

Arun Dayal Udai, C.G Rajeevlochana, Subir Kumar Saha

Abstract

The paper discusses a method for the dynamic simulation of a KUKA KR5 industrial robot using MATLAB SimMechanics. As it is difficult to model complicated 3D objects directly in SimMechanics environment, Autodesk Inventor was used to develop CAD model of KUKA KR5 robot parts and its assembly. Details of the CAD assembly and importing to the MATLAB environment are explained. The assembly of robot required joint constraints that were fed with cycloidal joint trajectories to perform inverse dynamics. The same system was also allowed to fall freely under gravity for which simulation results were obtained. The above inverse dynamics and simulation results were verified using those obtained from the model developed.

Keywords: CAD Assembly, MATLAB SimMechanics, Dynamic Simulation

1 Introduction

To verify the performance of a robotic system we normally perform computer simulation prior to actual purchase or fabrication. Analyzing a robotic system requires software packages that can perform mathematics, control, data acquisition, communication with external devices, image processing, real time data manipulation, 3D visualization, etc., and also programming support under a single platform. Graphical programming language is always an intuitive way to implement robotic simulation. There are packages like LabVIEW[1], MATLAB Simulink[2], SimMechanics[3], [4] and others [5] which support such an environment for programming and visualization. None of the literature surveyed so far reported how to integrate a CAD model of a robot to a simulation software environment to study its behaviour. Hence, the dynamic simulation of a KUKA KR5 industrial manipulator was carried out here to observe its behaviour and check its performances as it was planned to be procured for some real life applications. This forms a major contribution of the paper. MATLAB SimMechanics has been used to perform the simulation whose results are validated with those obtained using another commercially available software Autodesk Inventor.

Virtual simulations being a demanding field in robotics, several attempts have been made by the researchers to create and simulate robots in virtual environments. Different MATLAB-based tools are discussed in [6] with reference to the simulation of a robot manipulator, and its associated advantages and disadvantages. A benchmark model for testing various virtual simulation platform is discussed in [7]. It demonstrates an industrial robot using MATLAB SimMechanics. Many parallel manipulators were also simulated in using MATLAB SimMechanics [8], [9].

¹Arun Dayal Udai, ²C. G. Rajeevlochana, ³Subir Kumar Saha
Department of Mechanical Engineering, IIT Delhi, Hauz Khas, New Delhi, India
E-mail: ¹arun_udai@yahoo.com, ²lochana94@gmail.com, ³saha@mech.iitd.ac.in.

MATLAB SimMechanics is an efficient tool for teaching the design aspects of robotic manipulators [8]. The use of a CAD platform in conjunction with MATLAB SimMechanics was demonstrated in [10]. The paper uses SolidWorks for CAD Model development of a robot and SimMechanics was used to simulate the robot in virtual environment. Apart from pure mechanical system, SimMechanics was demonstrated in [11] as a hybrid simulation platform where an adaptive controller model was coupled with mechanical system.

In this paper, the 3D CAD parts of the KUKA KR5 [12] were assembled in Autodesk Inventor followed by the constraints. The robot's assembly was then exported as an XML and STL files using SimMechanics CAD translator module installed as an add-on to Autodesk Inventor. They were then imported in MATLAB SimMechanics as a model file. The imported model was attached with sensors and actuator controls to analyze the simulation for a cycloidal joint trajectories at each joint of the robot [13]. Similar procedure of importing and assigning assembly constraints was done in Autodesk Inventor and dynamic simulation was performed for the same joint inputs to validate the results of SimMechanics.

The robot model developed using SimMechanics will not only help in understanding the dynamic behaviour of KUKA KR5 but also allow one to integrate data acquisition, control, etc. to generate a virtual robot. Such virtual robot in turn may also be used as a training tool if connected with an external device like the robot's teach pendant.

The paper is organized as follows: Section 2 explains the CAD modelling of different parts of KUKA KR5 and their exporting, while section 3 explains the import procedure and assembly of the robot parts. Section 4 explains the simulation results followed by the conclusions in section 5.

2 CAD Modeling and Exporting

The CAD parts of KUKA KR5 industrial manipulator was obtained from KUKA's website [6], which were imported into the Autodesk Inventor platform. The parts were then assembled using joint constraints, and attached with the Denavit-Hartenberg (DH) frames [14] typically used to describe a robot's architecture. Based on the total mass of the robot as obtained from the specifications of KUKA KR5 robot and volumes as obtained from the CAD information using Inventor, individual masses of the parts were assigned assuming uniform density of the parts. Inventor automatically calculates the moment of inertias of each part about its centre of mass along the body axes. Information window of the Inventor showing mass properties of a link and the assembled model with DH frames attached is shown in Fig. (1). Masses and moments of inertia of all the six links are shown in Table 1. Table 2 shows the DH parameters for KUKA KR5 robot.

Using the CAD translator of SimMechanics [15] the KUKA KR5 model was converted to a format that can be imported to MATLAB Simulink environment. It requires to be installed from within MATLAB's command window using `install_addon` command. This creates the add-on toolbox in the main menu of the CAD software package, from where CAD model can be exported as a bundle of STL files and XML file. The STL file contains the CAD part definition of the links and XML file contains the topological information and physical properties of the robot

assembly. The XML file along with the STL files are then ready for importing into the MATLAB environment.

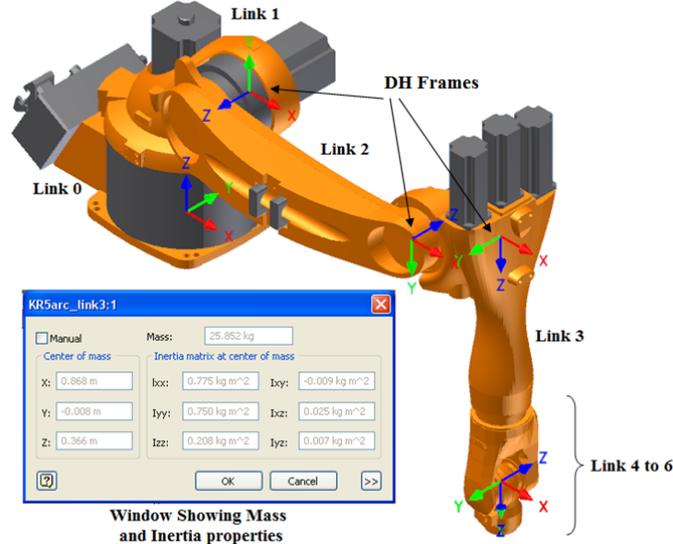


Figure 1: KUKA KR5 robot in home position.

Table 1: Mass and Inertia properties of various parts of KUKA KR5

Links (Fig. 1)	Mass (Kg)	Moment of Inertia (Kgm ²)	Remarks
0	55.4025	0.858811, -0.0111649, -0.0995688; -0.0111649, 1.52724, -0.00250204; -0.0995688, -0.00250204, 1.30055	Base supporting the robot structure
1	26.9797	0.321807, -0.0175473, -0.145272; -0.0175473, 0.467086, -0.0136195; -0.145272, -0.0136195, 0.477758	Link 1
2	15.9204	0.54137, 0.000435555, -0.00455128; 0.000435555, 0.55181, 0.0174856; -0.00455128, 0.0174856, 0.0436346	Link 2
3	25.8524	0.775113, -0.00947011, 0.0247925; -0.00947011, 0.750399, 0.00744338; 0.0247925, 0.00744338, 0.207558	Link 3
4	4.08846	0.00950111, 0.00163357, -8.41256e-006; 0.00163357, 0.0201731, 2.69102e-006; -8.41256e-006, 2.69102e-006, 0.0241324	Link 4
5	1.61544	0.00151607, 0.000409305, 0; 0.000409305, 0.00355535, 0; 0.0, 0.0, 0.00358942	Link 5
6	0.0160103	6.20477e-006, 0.0, 0.0; 0.0, 3.01352e-006, 0.0; 0.0, 0.0, 3.29301e-006	Link 6 End Effector Attachment

Table 2: DH Parameters of KUKA KR5 Robot

Link i	Link Length (mm) - a_i	Link Twist (degrees) - α_i	Joint Offset (mm) - d_i	Joint Angle θ_i
0	180	90	400	θ_0
1	600	180	135	θ_1
2	120	-90	135	θ_2
3	0	90	620	θ_3
4	0	-90	0	θ_4
5	0	0	115	θ_5

3 Importing CAD Assembly and Simulation

Once the parts are assembled on required plane and constraints were applied to the joints, the model was exported in the form of XML and STL files. The XML file was then imported in MATLAB environment by invoking `mech_import` command at the command prompt. The raw model with SimMechanics blocks comprising of different parts with their physical properties and appropriate joints was generated automatically after importing. The assembled parts that form a single link which move together were grouped to form a single block (subgroup) representing that link. Normally, a group of parts that move together were connected using weld joint block after importing. One must be careful while assembling and fixing constraints to the CAD parts in a CAD platform. SimMechanics translator exports only a definite set of joints. It places a revolute joint in place of any joint axis where one degree of freedom of rotation exists. However, the joints may be represented by a single axis motor, pin joint or a cylindrical joint in the CAD platform. Completely constrained part assemblies were imported as a group of parts connected with weld joints in SimMechanics.

The joints were then attached with joint actuator block which inputs the cycloidal motion trajectory [13] to the joint given by

$$\theta_i(t) = \theta_i(0) + \frac{\theta_i(T) - \theta_i(0)}{T} \left[t - \frac{T}{2\pi} \sin\left(\frac{2\pi}{T}t\right) \right] \quad (1)$$

where θ_i is the joint variable and T is the total time of the trajectory. Corresponding velocity and acceleration are given by

$$\dot{\theta}_i(t) = \frac{\theta_i(T) - \theta_i(0)}{T} \left[1 - \cos\left(\frac{2\pi}{T}t\right) \right] \quad (2)$$

$$\text{and } \ddot{\theta}_i(t) = \frac{\theta_i(T) - \theta_i(0)}{T} \left[\frac{2\pi}{T} \sin\left(\frac{2\pi}{T}t\right) \right] \quad (3)$$

Cycloidal trajectory ensures zero starting and ending joint velocities and accelerations whose expressions are given in Eq. (2) and Eq. (3), respectively. Comparing Eqs. (1) – (3), the acceleration expression is the most simple in mathematical form. Its expression was used as a function for trajectory input. For one second of simulation a sinusoidal function generator block with frequency and amplitude of 2π can be used. The acceleration was then integrated twice with initial and final conditions imposed as boundary values. Each joint actuator takes multiplexed vector form of the joint position, velocity and acceleration to actuate a

particular joint. The representative block layout for the entire process is shown in Fig. (2), which shows links 0, link 1 and link 2 with actuators.

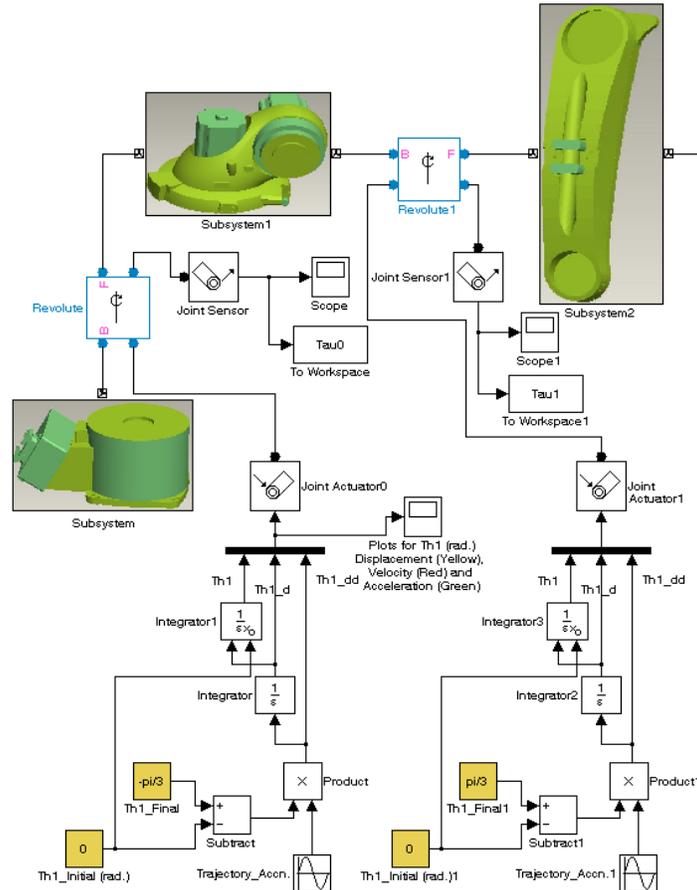
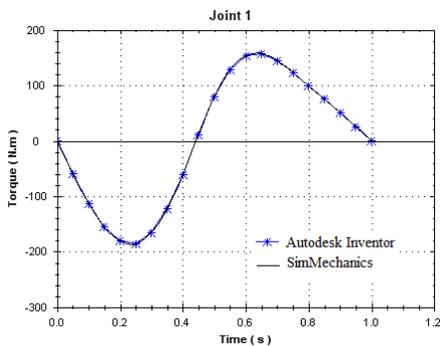


Figure 2: First two joints of the Simulation Layout in MATLAB SimMechanics

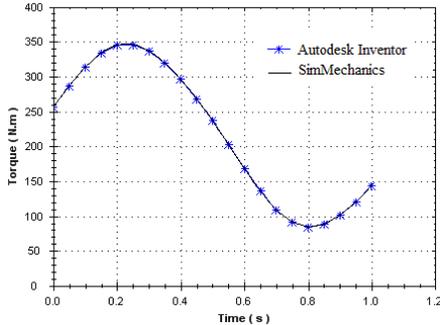
4 Dynamic Simulation Results

The joint blocks were attached with the joint sensor of MATLAB SimMechanics that can be opted to sense the computed torque of the revolute joints in Nm. Inverse dynamics, i.e., to obtain the joint torques to follow the specified trajectories, Eqs. (1) – (3), $T = 1$ sec. was performed whose plots are also shown in Fig. (3). Note that the results of only first three joints are shown due to space limitations. Similar trajectories were given to the model built on Autodesk Inventor environment as well. The torques obtained are also shown in Fig. (3). It was observed that the torques obtained by both the software are in close agreement to each other. Hence, the results are correct. The SimMechanics model can now be used for further control and other numerical experimentations for better understanding of the robot's behaviour.

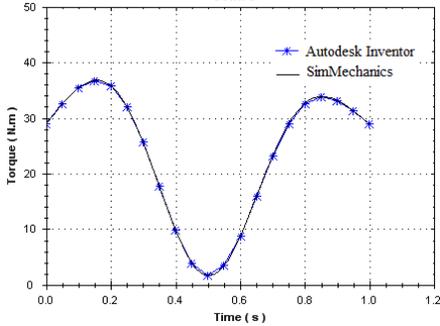
In order to test how the robot would behave without any actuator torques, simulation was performed by delinking the joint actuators, i.e., no joint torques to be applied. The robot was left free to fall under gravity. The joint angles were obtained by joint sensors which measures the variation in radians with time. These results are shown in Fig. (4). To see how the end-effector's centre of mass of the last link, i.e., the end-effector will move, body sensor was attached to it and the plots were obtained, as shown in Fig. (5). The simulated results from SimMechanics were also verified with those obtained using Autodesk Inventor. The overall process is shown through block diagram in Fig. (6).



(a) Joint 1

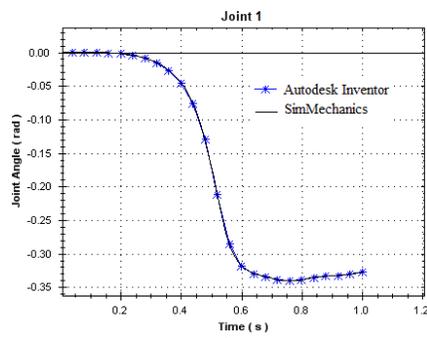


(b) Joint 2

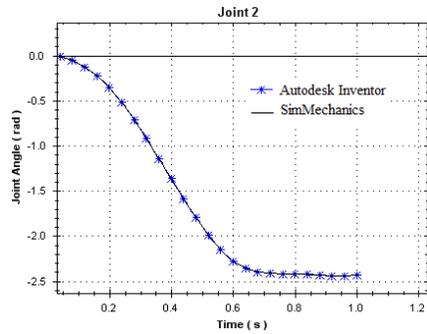


(c) Joint 3

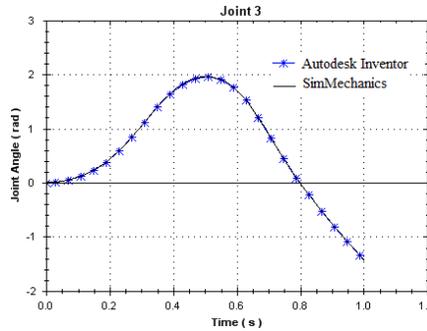
Figure 3: Joint torques



(a) Joint 1



(b) Joint 2



(c) Joint 3

Figure 4: Free fall simulation

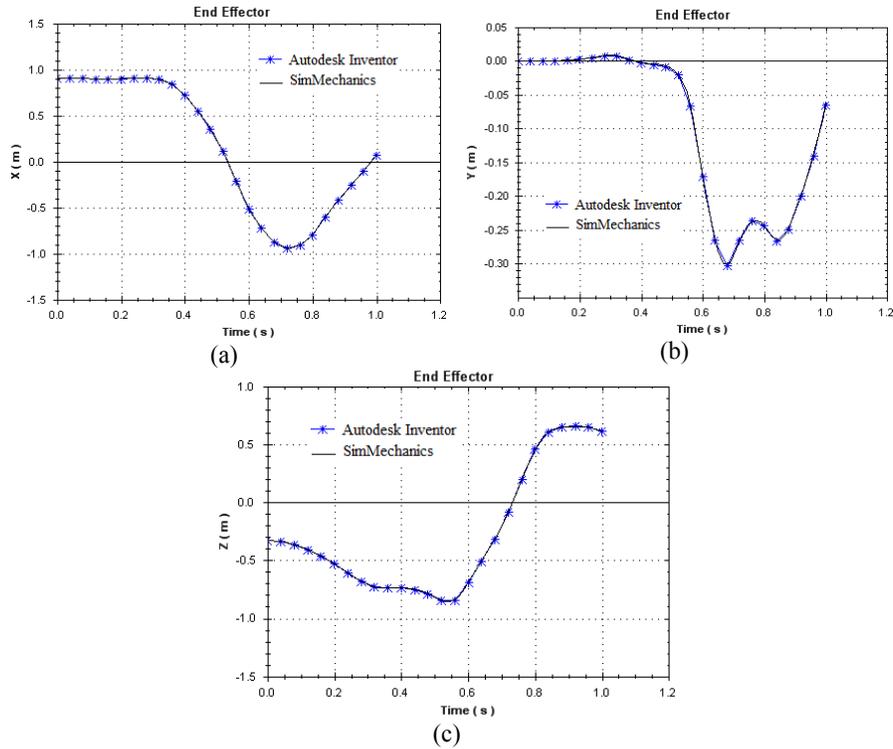


Figure 5: Plots for end-effector's position

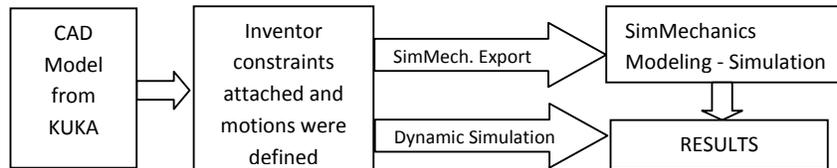


Figure 6: Block diagram of the overall process.

5 Conclusions

Graphical programming languages like MATLAB SimMechanics or Autodesk Inventor can be utilized as powerful tools for simulating a robotic system as it is reported in this paper. However, SimMechanics is recommended here as it can accommodate control laws, data acquisition systems, etc. to have almost a full-fledged integrated robotic system in virtual world, whereas Autodesk Inventor is primarily a CAD/CAE tool which does not have the features required for complete robotic simulation. Hence, the present work is the first step towards an integrated robot simulation for the KUKA KR5 robot which the authors have procured to use for some real-life applications like peg-in-hole insertion. The SimMechanics is

planned to be used with control etc. in virtual environment first before the real control is done with the real KUKA KR5 robot.

Acknowledgement

The research work reported here was carried out under the project entitled “Adaptive Force Control of an Industrial Robot (KUKA KR6) equipped with Force/Torque sensor,” which is one of the five projects for “Program for Autonomous Robotics” sponsored by Board for Research in Nuclear Science (BRNS). The financial support provided to the first two authors from the project is sincerely acknowledged.

References

- [1] S. S. Sun, and Q. H. M. Meng, “Dynamic Simulation of Robot Manipulators Using Graphical Programming Packages,” Proc. of IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing, Victoria, BC, Canada, 1995, pp: 521-524.
- [2] SIMULINK User’s Guide, The Mathworks, Inc., 1992
- [3] G. D. Wood, and D. C. Kennedy, “Simulating Mechanical Systems in Simulink with SimMechanics,” Tech. Report, The Mathworks Inc., 2003.
- [4] M. Schlotter, “Multibody System Simulation with SimMechanics,” Technical Report, Darmstadt University of Technology, 2003.
- [5] M. S. Alshamasin, F. Ionescu, and R. T. Al-Kasasbeh, ‘Kinematic Modeling and Simulation of a SCARA Robot by Using Solid Dynamics and Verification by MATLAB/Simulink,’ European Journal of Scientific Research, Vol.37 No.3 (2009), pp.388-405.
- [6] L. Zlajpah, ‘Simulation in robotics,’ Proceedings of Mathematics and Computers in Simulation, Elsevier, 2008, pp. 879–897
- [7] J. Franke, and M. Otter, “The Manutec r3 Benchmark Models for Dynamic Simulation of Robots”, DLR FF-DR-ER, Technical Report TR T101-93, March 1993.
- [8] N. S. Tlale, and P. Zhang, “Teaching the design of parallel manipulators and their controllers, implementing MATLAB, Simulink, SimMechanics and CAD,” Int. J. Engng. Ed. 21(5), 2005, pp: 838-845.
- [9] L. Yu, L. Zhang, and S. Yang, “Kinematic simulation and analysis of 3-RPS parallel robot on SimMechanics,” Proc. Int. Conf. On Information and Automation, Harbin, China, June 2010, pp: 2363-2367.
- [10] M. I. C. Dede, and S. Tosunoglu, “Virtual rapid robot prototyping,” ASME Early Career Technical Journal 5(1), 2006, pp: 7.1-7.8.
- [11] I. Hassanzadeh, A. Harifi, and F. Arvani, “Design and Implementation of an Adaptive Control for a Robot,” American J. of Appl. Sc. 4(2), 2007, pp: 56-59.
- [12] KUKA Robotics website <http://www.kuka-robotics.com/usa/en/downloads/>
- [13] S. K. Saha, *Introduction to Robotics*, New Delhi: Tata McGraw Hill, 2008.
- [14] J. Denavit, and R. S. Hartenberg, “A kinematic notation for lower-pair mechanisms based on matrices,” Trans. ASME J. Appl. Mech (23), pp: 215-221, 1955.
- [15] SimMechanics CAD Translator, available at http://www.mathworks.com/products/simmechanics/download_cad.html.