Kinematic Analysis and Simulation of a 6 DOF Robot in a Web-Based Platform Using CAD File Import

Ujjal Dey(✉) and Kumar Cheruvu Siva

Department of Mechanical Engineering, Indian Institute of Technology (IIT), Kharagpur 721302, India
ujjal.dey@iitkgp.ac.in, kumar@mech.iitkgp.ernet.in

Abstract. The current trend of simulator-based analysis, especially in the area of robotics had emerged broadly. This kind of simulation gives initial familiarization of the system, which is very useful for introductory level courses in robotics as well as research-based work. Simulation developed through any commercial software's requires its installation on the user's system for any animation or analysis. Therefore, an open source platform for this type of robot motion analysis had a much impact due to its light version, better graphics, and web-based running capability. This paper describes an efficient and very straightforward approach of building a 6 degree of freedom KGP50 robot simulation model in a web interface using WebGL technology. Here a component is first designed in SolidWorks and then it is imported directly into the WebGL-based platform utilizing a library of Three.js. The forward kinematics analysis of our KGP50 robot is presented through this simulator, which gives the idea of the whole framework and also an exploration of the KGP50 robot.

Keywords: KGP50 · WebGL · Web · Three.js · CAD model

1 Introduction

Analysis of any sort of robot motion through simulation requires creating a virtual model of its representation. In general, a 3D model is build using some CAD designing software and its simulation is observed through MATLAB/Simulink/Simscape based environment [1–5]. Some other tools like RoboAnalyzer, RoboDK are also available for this type of kinematic study and programming of robots [6]. This procedure requires a user to have that software installed in the system to run the simulation, which in many cases may put a constraint for frequent access. Hence, web-based interfaces are developed to visualize 3D simulations with interactive application enabled, which is free from any supporting software or plugins. For better enrichment of the display of 3D-content WebGL based platform are emerging. Web Graphics Library, (WebGL) is a JavaScript Application Programming Interface (API) for rendering 2D or 3D graphics within any HTML5 supported web browser [7]. It uses the HTML5 canvas element and is accessed using Document Object Model (DOM) interfaces.

A. Abraham et al. (Eds.): ISDA 2017, AISC 736, pp. 243–250, 2018.
https://doi.org/10.1007/978-3-319-76348-4_24

WebGL provides creating 3D geometries with specific commands from which a virtual model is made, and its simulation is performed. For complicated 3D components method of CAD file import is used for building robot models. A 3D CAD model is first designed in software's like Solidworks, CATIA, Creo, etc. and then extracted triangular vertex coordinate data from the file is implied into the program. This type of WebGL technology method has been implemented with various approaches for standard industrial robot motion simulations [8]. It seems to be a fine approach, but the direct implication of WebGL technology is very tiresome, especially when a model with multiple components is to be rendered. For viewing a 3D element one need to describe the entire vertex coordinates extracted from the CAD file through some application. Therefore, using a library reduces the steps and makes the process much easier and faster. Currently, many JavaScript libraries like Three.js, SceneJS, BabylonJS uses WebGL and provide many high-level features for 3D object visualization. Three.js is a high-level utility library which makes WebGL program much easier and more straightforward. Since it uses JavaScript coding language, it can be combined with other libraries, which are capable of giving interactive applications. It also has different features of camera control, light control, and animations. Implementation of this technology for various web-based experimentations is going on in many fields of study [9, 10]. This paper describes the simulation and forward kinematic analysis of the KGP50 robot and also its workspace visualization in a web-based platform. The virtual model of the robot is created through the direct import of CAD file in STL format. Here WebGL technology is explored via Three.js library for building the simulator, which gives the idea of the whole framework of importing a particular CAD file, and its assemblage with other components.

2 The Methodology of Importing a CAD Model

Three.js is an open source JavaScript 3D library which uses WebGL Technology [11, 12]. It is used to create and display 3D scene through WebGL renderer module. For viewing a particular object, one needs to set up a Canvas within which a scene is created and position of the camera is specified. Then that scene is then rendered through THREE.WebGLRenderer() function. Three.js has some predefined commands for creating some basic 3D geometries, which is appropriately assembled to model some simple robotic configuration [13]. Since KGP50 has an intricate design, therefore methodology of CAD file import is needed for accurate modeling and better visualization. A particular part is first designed in SolidWorks and saved as stereolithography (STL) format. Then after preparing the primary interface of the program, it allows a user to visualize and manipulate that STL geometry within HTML canvas. In order to view the 3D model, the path of CAD file location and STL loader library must be specified properly within the program. The flow diagram of the framework for viewing a CAD model within the scene using Three.js library is given in Fig. 1. Apart from STL format Three.js also support other extensions of the 3D object, which can be loaded through that particular loader library. One can add different material and texture to make it more visually compelling.

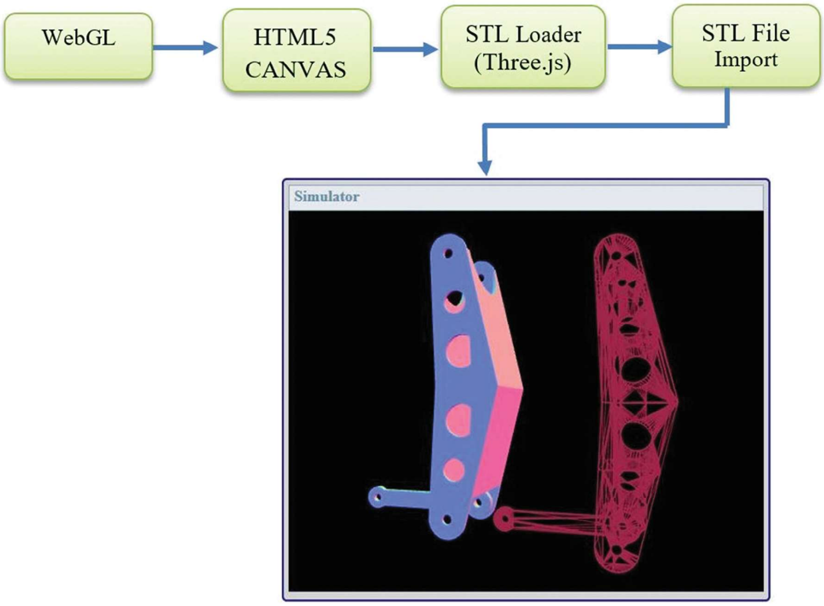

Fig. 1. Sequential steps for viewing a CAD model within canvas

3 KGP50 Robot

KGP50 is a prototype industrial robot with all features of a manipulator along with modular controllers and control techniques. It has six degrees of freedom with high payload capability and parallel linkage structure with the recirculating screwball

Fig. 2. KGP50 model

mechanism in joints 2 and 3. It is used as a testbed for cutting-edge technologies of human-computer interaction and intelligent systems. The robot is powered by digital servo control of AC motors synchronized and precision-controlled for coordinated motions using a real-time digital control station, working on a Digital Signal Processing-based motion controller. The actual model of the KGP50 robot is shown in Fig. 2.

Specifications:

- 6-Axis, Continuous path control
- 50 kg Payload
- 1.5 m reach
- 1.5 m/s maximum speed
- 0.1 mm repeatability

3.1 Kinematics

Kinematics is the science of motion that treats the subject without regard to the forces that cause it. In robot kinematics, two major aspects of the study are the forward and inverse kinematics analysis of any manipulator. The problem of forward kinematics deals with the determination of the position and orientation of the end-effector of the manipulator for a given set of joint angles. Here forward kinematics analysis is carried out by assigning D-H parameters to all links. To assign the D-H parameters kinematic diagram of the robot is the first setup, which is presented in Fig. 3. D-H parameters of the robotic arm are given in Table 1. The transformation matrix relation between the end effector and the base frame attached to the robot base is expressed as:

$$ {}^{0}\mathrm{T}_{6} = {}^{0}\mathrm{T}_{1}\,{}^{1}\mathrm{T}_{2}\,{}^{2}\mathrm{T}_{3}\,{}^{3}\mathrm{T}_{4}\,{}^{4}\mathrm{T}_{5}\,{}^{5}\mathrm{T}_{6} \tag{1} $$

Each homogeneous transformation matrix is expressed as a product of four basic transformations associated with joints i and j (l-link length, α-link twist, d-link offset, and θ-joint angle) and I is a 4×4 identity matrix. The general form of each transformation matrix of the i^{th} frame with respect to $i-1^{\mathrm{th}}$ frame is given by [14]:

$$ {}^{i-1}T_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1} d_i \\ S\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

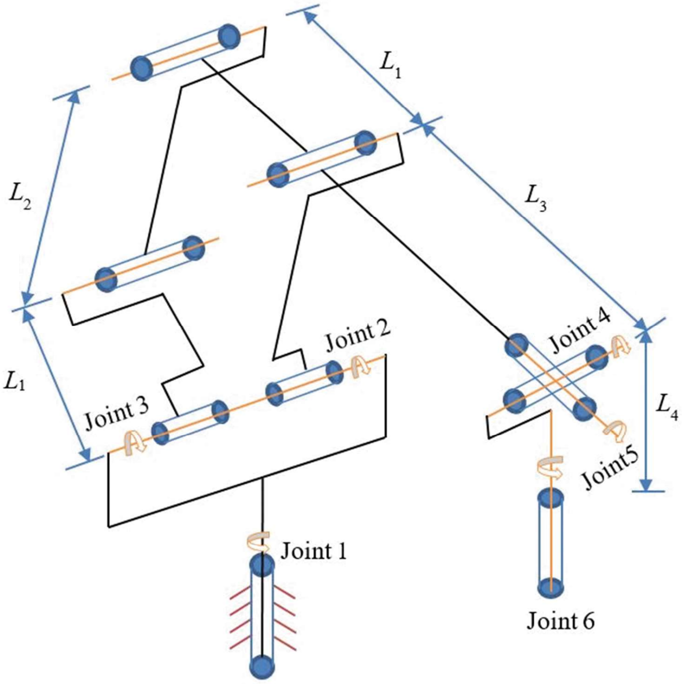

Fig. 3. Kinematic diagram

Table 1. D-H parameters of KGP50

Link i	α_{i-1}	a_{i-1} (mm)	d_i (mm)	θ_i
1	0	0	0	θ_1
2	−90	0	0	θ_2
3	0	L_2	0	θ_3
4	−90	0	L_3	θ_4
5	90	0	0	θ_5
6	−90	0	L_4	θ_6

4 Assembly and Simulation of KGP50 Robot

To perform the kinematic analysis, the whole model is constructed by importing all the components of KGP50 and assembling them sequentially. As described earlier, once an STL file is introduced within the HTML5 canvas it is considered as geometry, and that geometry can be manipulated as desired. It can be translated and rotated any amount as desired within the canvas for assembling. For accurate positioning and orientation of the geometry, the transformation matrix can be applied to that geometry. The assembled virtual model of KGP50 is shown in Fig. 4. The whole model can be visually enhanced by adding different materials and texture, which will look almost similar to our actual robot.

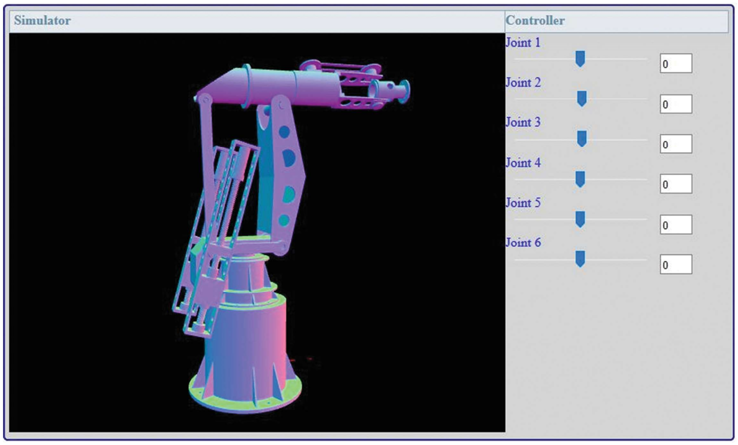

Fig. 4. Assembled 3D virtual model of KGP50

After the model is ready, separate programs are developed to observe it's all joint motions separately and perform forward kinematic analysis. An Input panel is created for submission of joint angles data. Specifying the joint angular values final transformation matrix is displayed, which shows the position and orientation of the tool frame with respect to the base frame. In robot kinematics, one other important aspect of the study is the workspace analysis of robot, which indicates the volume of space that the end-effector of the manipulator can reach. Therefore a charting library of plotly.js is used for graphical representation of manipulator position information. For each submission of joint angular values, a 3D scatter graph is generated which indicates end-effector position and from which workspace of the robot can be visualized. Now switching the format of the chart from point to line graph the motion trajectory of the robot can be seen. Figure 5 shows two different configurations of the KGP50 robot in the forward

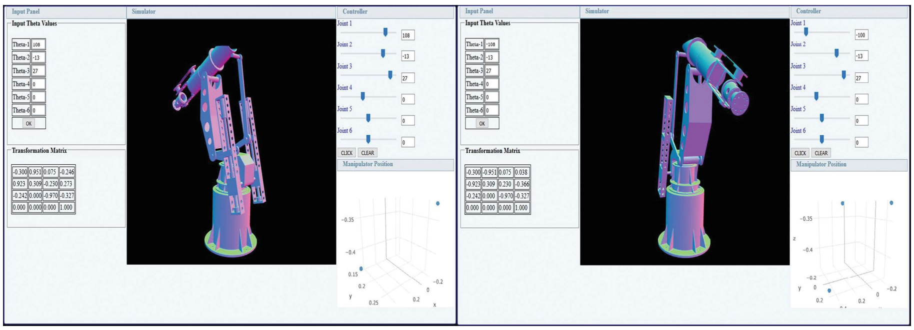

Fig. 5. Screenshot of the interface for two distinct configurations of the robot

kinematics analysis. The whole program in compacted within an HTML file, which can be run in any WebGL compatible web browser for analysis.

5 Conclusion

This paper illustrates the utilization of Three.js library for developing the 3D model of the KGP50 robot which is used for its forward kinematic analysis in a web-based virtual environment. It uses the technology of WebGL, which provides a very efficient platform for 3D object visualization. Using the library of Three.js helps in the direct rendering of the 3D object in Cartesian space. Any 3D model can be easily built up using this CAD file import procedure, which removes commercial software's dependencies. The same environment can be further used for Inverse kinematics analysis by just adding the mathematical calculation in a JavaScript file format. Here multiple robots motion simulation can also be incorporated into the same environment, which is a unique feature in this type of methodology. This kind of web-based platform is beneficial for virtual laboratories, which enhance the process of E-learning/education. One major prospect of this kind of open-source program is that the source code is freely available to others and it can be modified as desired to improvise the design. Since the whole analysis is carried out considering actual dimensions and kinematic parameters, it is almost similar to real system operation. It can be further programmed to connect this simulation model and real-time control system of the actual robot for remote operation.

Acknowledgement. The authors gratefully acknowledge the financial support from MHRD, New Delhi and Department of Mechanical Engineering, Indian Institute of Technology, Kharagpur for providing the platform to carry out the work.

References

1. Alshamasin, M.S., Ionescu, F., Al-Kasasbeh, R.T.: Kinematic modeling and simulation of a scara robot by using solid dynamics and verification by Matlab/Simulink. Eur. J. Sci. Res. **37**(3), 388–405 (2009)
2. Gousmi, M., Ouali, M., Fernini, B., Meghatria, M.H.: Kinematic modelling and simulation of a 2-R robot using SolidWorks and verification by MATLAB/Simulink. Int. J. Adv. Robot. Syst. **9**(6), 245 (2012)
3. Zodey, S., Pradhan, S.K.: Matlab toolbox for kinematic analysis and simulation of dexterous robotic grippers. Procedia Eng. **97**, 1886–1895 (2014)
4. Corke, P.: A robotics toolbox for MATLAB. IEEE Robot. Autom. Mag. **3**(1), 24–32 (1996)
5. Fedák, V., Ďurovský, F., Üveges, R.: Analysis of robotic system motion in SimMechanics and MATLAB GUI environment. In: Bennett, K. (ed.) MATLAB Applications for the Practical Engineer, 3rd edn. InTech (2014)
6. Bahuguna, J., Chittawadigi, R.G., Saha, S.K.: Teaching and learning of robot kinematics using RoboAnalyzer software. In: Proceedings of First International Conference on Advances in Robotics, Pune, pp. 1–6. ACM (2013)
7. Tavares, G.: WebGL fundamentals. HTML5 Rocks (2012)

8. Lianzhong, L., Kun, Z., Yang, X.: A cloud-based framework for robot simulation using WebGL. In: Proceedings of Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA), Marrakesh, pp. 5–8. IEEE (2015)
9. Severa, O., Goubej, M., Konigsmarkova, J.: Unified framework for generation of 3D web visualization for mechatronic systems. J. Phys. Conf. Ser. **659**(1), 012053 (2015)
10. Tudjarov, B., Botzheim, J., Kubota, N.: Facilitation of cognitive robotics by web-based computational intelligent models. In: Micro-NanoMechatronics and Human Science (MHS), pp. 144–148. IEEE (2012)
11. Dirksen, J.: Learning Three.js: The JavaScript 3D Library for WebGL. Packt Publishing Ltd., Birmingham (2013)
12. Parisi, T.: Programming 3D Applications with HTML5 and WebGL: 3D Animation and Visualization for Web Pages. O'Reilly Media, Inc., Sebastopol (2014)
13. Dey, U., Jana, P.K., Kumar, C.S.: Modeling and kinematic analysis of industrial robots in WebGL interface. In: Proceedings of Eighth International Conference on Technology for Education (T4E), Mumbai, pp. 256–257. IEEE (2016)
14. Craig, J.J.: Introduction to Robotics: Mechanics and Control. Pearson Prentice Hall, Upper Saddle River (2005)