Design and Development of a Web-Based Robotics Simulator

Amit Kumar Nandi and Laxmi Narayan Pattanaik

March 19, 2020

Design and Development of a Web-Based Robotics Simulator

Amit Kumar Nandi
Department of Poduction Engineering
Birla Institue of Technology, Mesra
Ranchi, India
ORCID: 0000-0002-2626-1351

Dr. Laxmi Narayan Pattanaik
Department of Poduction Engineering
Birla Institue of Technology, Mesra
Ranchi, India
lnpattanaik@bitmesra.ac.in

***Abstract*—On the advent of Industry 4.0, there is an increasing demand in software for robot modelling and simulation. Most currently available solutions runs natively on the user system which cuts down on the availability and also takes a toll on the computing resources. The idea of this paper is to develop a novel robotics simulation software that runs out-of-the-box in a web-browser. It doesn't need installations and high computing resources. Any machine capable of running a web-browser can test and use this simulation software. Major objective remains to target the didactics and educational market where students can learn and validate kinematic equations visually by running them directly on a robot or on a simulated version of the same.[6]**

Keywords—robotics, simulation, kinematics, modelling, didactics

I. Introduction

With increasing demand of simulation software for the robotics domain in industry, we have come up with a novel solution for modelling and learning robot kinematics that will need the least amount of pre-requisites to get the system up and running. Existing solutions require installing lot of dependencies and a basic amount of IT proficiency is required to get them running in a system while our solution requires only a web-browser to get it running.[3] Developed by keeping in mind the ease of use and also powerful modelling manipulation function, we have an overwhelming response from the users via our hosted solution in the internet which can be used right away through this URL: https://nandimechatronics.me/rg/robosim.html. Existing robot market forecast says it will grow exponentially in the coming five years in terms of sales . Some of the supporting data are given below in graphical format

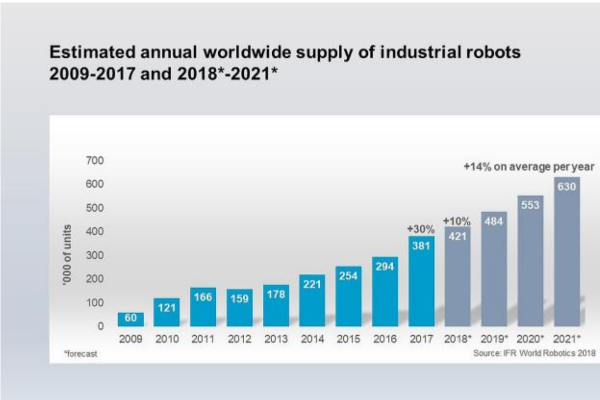

Figure 1: Worldwide robotics sales chart(Yearly)

Figure 2: Robotics Market Overview

II. Prposed Model

A. Brief

The software solution is developed wholly in the Javascript language.[1] We are using quite a number of open- source libraries for graphics, kinematics modelling and also remote procedure calls. Our software itself is open sourced on GitHub and can be modified or used right away by anyone who is interested for the same.[3] The major parts of this solution is the Graphical modelling, the mathematical modelling and the front-end design framework.

B. Major Components

Some of the major steps taken to develop the solution is given below in the bulleted list form:

1. Graphical Modelling
2. Kinematics Modelling
3. Movement Behavior Modelling
4. Coding
5. Testing
6. Hosting and Serving to users

III. METHODOLOGY

The graphical implementation of software is done using OpenGL complaint Javascript library known as ThreeJS.. The kinematics and behavioral logic is written on plain Javascript only. [1] The presentation components is developed on HTML Document Object Model. Some of the major features provided in our solution are given below:

1. Defining Model Geometry Parameters (Joint Type, No of Joints, Link Type, No of Links, Link Length)
2. Translational/Rotational Movement
3. Euler Angle representation in the model space
4. Forward/Inverse Kinematics simulation
5. Path Planning
6. Possible point positional representation

7. Real-Time angle joint-angle representations
8. On-Line Simulation on real machines and code verification
9. Export and Import of Model/Code

A. *Functional Block Diagram*

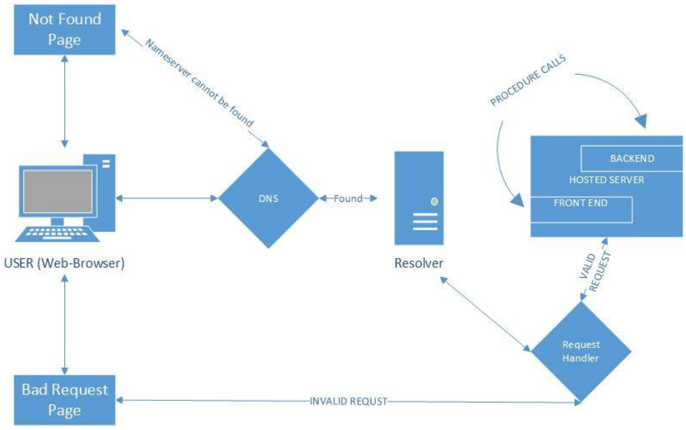

Figure 3: functional block diagram of the working solution

B. *In-Depth Funtionalities*

- In the forward/inverse kinematic simulation, user can get joint angle or end-effector positions for any configuration of robot given that link-lengths are conformational to Denavit-Hartenberg parameters and the joint angles are conformational to the Euler Quaternion parameters
- The point-space mappings are done to the accuracy of 0.05 cm per point
- The model export and import is done by translating the same into our intermediatory XML schema that is transportable and supported by a lot of platforms in the robotics domain. Though limited interoperability can be achieved till now.
- The solution is platform and hardware agnostic and can be run in fully functional mode in systems having support for just a V8 Javascript engine complaint web-browser

C. *Equations*

General solutions for the inverse and forward kinematic models have been implemented in the software with highest no of degrees of freedom supported by the system being six and minimum being 2. The following equations have been implemented in the component behavioural logic of our solution:[4]

$$\frac{\partial p_i}{\partial x_k} \approx \frac{p_i(x_{0,k}+h) - p_i(x_0)}{h}, \quad \text{--- (1)}$$

$$[T] = {}^0T_n = \prod_{i=1}^{n} {}^{i-1}T_i(\theta_i), \quad \text{---(2)}$$

Identify applicable funding agency here. If none, delete this text box.

The D-H general matrix is taken as follows [7]:

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_{i,i+1} & \sin\theta_i\sin\alpha_{i,i+1} & a_{i,i+1}\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_{i,i+1} & -\cos\theta_i\sin\alpha_{i,i+1} & a_{i,i+1}\sin\theta_i \\ 0 & \sin\alpha_{i,i+1} & \cos\alpha_{i,i+1} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

D. *Open-Source Libraries Used*

- **ThreeJs**: Three.js is a cross-browser JavaScript library and application programming interface (API) used to create and display animated 3D computer graphics in a web browser.[8] Three.js uses WebGL. The source code is hosted in a repository on GitHub.Three.js allows the creation of graphical processing unit (GPU)-accelerated 3D animations using the JavaScript language as part of a website without relying on proprietary browser plugins. This is possible due to the advent of WebGL.
- **WebGL**: WebGL (Web Graphics Library) is a JavaScript API for rendering interactive 2D and 3D graphics within any compatible web browser without the use of plug-ins It is fully integrated with other web standards, allowing GPU-accelerated usage of physics and image processing and effects as part of the web page canvas. WebGL elements can be mixed with other HTML elements and composited with other parts of the page or page background. WebGL programs consist of control code written in JavaScript and shader code that is written in OpenGL ES Shading Language (GLSL ES), a language similar to C or C++, and is executed on a computer's graphics processing unit (GPU). [5]
- **Bootstrap**: Bootstrap is a web framework that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight. Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container. While

the latter always fills the width of the web page, the former uses one of the four predefined fixed widths, depending on the size of the screen showing the page:

IV. Result and Dicussions

The software has been completely developed and tested by me and my project guide and after several deign iterations its finally been hosted to the internet for public use. The response since then has been overwhelming after we have reached nearly a lakh of concurrent users for our system. A community formation is in the works for constant feedback collection and support to our users which will make the product even better in the upcoming days.[4]

A. Availability to Users

The free version of our software is currently available for public use in the following URL: http://nandimechatronics.me/rg/robosim.html

Enterprise version is currently in development with lot of improvements done directly based on user feedback and A/B testing and soak testing our iterative solutions ad the same will soon be available on demand from our website

B. Limitations

Currently the software solution only supports a maximum of six degrees of freedom configurations and also the working space mappings' accuracy is capped to 0.05 cm per point to point map. Constant optimization is being done to remove and overcome such limitations in the software with active collaboration and user feedback summarizations.[2]

Online simulation on real robots are currently limited to robots supporting Karel Programming Language like some models of FANUC and KUKA robots.[2] So this makes the online simulation interoperability limited to the supported models.

C. Related work in context of the proposed system

A few initiatives has been taken previously by researchers and academicians around to world to put together a system of robotics simulation that runs out-of-the-box in a web-browser. Most of the work has been done before there was AJAX and thus the interactivity of such systems was limited and no real-time output was provided. Some of the works related to the proposed system that has been done previously and worthy of mention have been discussed briefly in a bulleted list below:

- A web-based robotic simulator in Java [9]: Here in this initiative, Java 3D has been used to draw and simulate a multi-linked robot kinematics using the VRML robot programming syntax. It has some basic features like jogging, control and advanced features like collision detection and avoidance for optimal path planning. Given that the application was written in Java , browser must support web-applets to run the same.
- The Player Project is an umbrella under which two robotics-related software projects are currently developed. These include the Player networked robotics server, and the Stage 2D robot simulation environment. The project was founded in 2000 by Brian Gerkey, Richard Vaughan and Andrew Howard at the University of Southern California at Los Angeles, and is widely used in robotics research and education. It has evolved to the name of Gazebo Simulator in modern days.[10]

D. Conclusion

The software has accurate representation of kinematic models and linkage behaviors though a lot of improvement needs to be done on verification on real machines in terms of interoperability. This project is the initial implementation of the idea that it is highly possible to simulate robot kinematics in the web-browser without the need for resource heavy software. More such initiatives are expected in the upcoming years to leverage the growth of Industry 4.0 via robotics.[5]

Acknowledgment

The completion of any project brings with it a sense of satisfaction, but it is never complete without thanking those people who made it possible and whose constant support has crowned our efforts with success. I would like to thank Prof (Dr.) Laxmi Narayan Pattanaik of Birla Institute of Technology, Mesra for letting us form a group and carry out the project and also for allowing us to use the college infrastructure to complete the project and test it. I hereby thank our respected Head of the Dept. of Production Engineering, Prof (Dr.) Sanjay Kumar Jha for providing constant inspiration. I also would like to thank all the faculty& staff members of Production Engineering Dept. for providing us with the required facilities and supports towards the completion of the project.

References

[1] Dong Kwon Cho, Sang Hun Chun, Jeong Gyun Ahn, Yong Sik Kwon, Jong Hoon Eom, and Young II Kim, 2009, A UDP-based protocol for mobile robot control over wireless Internet, 2nd International ICST Conference on Robot Communication and Coordination (ICST CRCC 2009).

[2] Wan Muhamad Hanif Wan Kadir, Reza Ezuan Samin, Babul Salam Kader Ibrahim, 2012, Internet Controlled Robotic Arm, International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).

[3] Meftah Zouai, Okba Kazar, Belgacem Haba, Guadalupe Ortiz, Nadia Kabachi, 2019, New approach using an IoT robot to oversight the smart home environment, ISTE OpenScience © – Published by ISTE Ltd. London, UK – (openscience.fr | 2019).

[4] Partha Pratim Ray, Internet of Robotic Things: Concept, Technologies, and Challenges, 2017, IEEE Access, (Vol. 4, 2017)

[5] Mário J. G. C. Mendes, Íris M. M. S. Neto, João M. F. Calado, 2018, Fault Diagnosis System Via Internet Applied to a Gantry Robot - A Proposal for Industry 4.0, 18th IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC April 25-27 2018)

[6] DingjuZhu, Cloud robot system and method of integrating the same, US Patent US9,031,692B2 (May12,2015)

[7] Cooper, S., Dann, W., Pausch, R.: Alice: a 3-D tool for introductory programming concepts. J. Comput. Sci. Coll. 15, 107–116 (2000). http://dl.acm.org/citation.cfm?id=364133.364161

[8] Threejs: Three.js (2020). https://threejs.org/

[9] Wu-Jeng LiWu-Jeng Li, A Web-based Robotic Design and Simulation System in Java 3D, The Second International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore (December 2003)

[10] Open Robotics Organization https://www.openrobotics.org/