

Using WebGL in Developing Interactive Virtual Laboratories for Distance Engineering Education

Dr. Mert Bal, Miami University

Mert Bal received his PhD degree in Mechanical Engineering from the Eastern Mediterranean University, North Cyprus in 2008. He was a Post-Doctoral Fellow in the University of Western Ontario, and a Visiting Researcher at the National Research Council Canada in London, Ontario, Canada between 2008 and 2010. He was involved in various research projects in the areas of collaborative intelligence, localization and collaborative information processing in wireless sensor networks, intelligent agents, agent-based manufacturing scheduling, systems control and automation, distributed control of holonic systems and integrated manufacturing, agile manufacturing, virtual reality and remote laboratory applications in education. He has authored or co-authored various journal and conference publications in these areas. Mert Bal is currently an Associate Professor in the Miami University, Department of Engineering Technology, Ohio, United States of America.

Dr. Reza Abrishambaf, Miami University

Reza Abrishambaf received his PhD degree in Electrical and Electronic Engineering from Eastern Mediterranean University, Cyprus in 2012. His research interests include Distributed Control Systems, Intelligent Manufacturing, Wireless Sensor Networks, Industrial Automation and IEC61499 Function Blocks. In 2012, he joined the University of Minho in Portugal and worked as an invited assistant professor and senior researcher in the department of industrial electronics and Centro Algoritmi of the same university where involved in several industrial and research projects in the field of Wireless Sensor Networks and Industrial Embedded Applications. He is currently an Assistant Professor with Department of Engineering Technology at Miami University, USA. Dr. Abrishambaf is an IEEE associate Member with more than 10 publications in the international journals and conferences.

Using WebGL in Developing Interactive Virtual Laboratories for Distance Engineering Education

Abstract

This paper investigates the use of WebGL (Web-based Graphics Library)-based interactive graphics technologies in realization of an immersive virtual reality (VR) platform for hands-on laboratory instruction in distance engineering education. Realizing the fact that the Web and 3D graphics are currently ubiquitous in desktop and mobile devices, we present a proof-of-concept approach that provides an inexpensive alternative of complex and costly immersive VR simulators. A prototype implementation of the proposed approach is also presented in order to demonstrate the convenience of VR platform in distance education in engineering technology.

1. Introduction

Distance learning is considered to be one of the most challenging areas in the field of engineering; students who have no or limited access to on-campus facilities, and little to no face-to-face contact with academic staff and fellow students face the risk of not being able to engage in course-work material and can feel isolated. Due to the size and the high cost of the laboratory equipment, it is often impractical and unaffordable for universities to provide hands-on laboratory experiences to their distance learners using actual laboratory equipment¹. Several programs have implemented solutions for this problem including the lab kits that contain small, inexpensive laboratory equipment that each student either purchases one or given one by the university in order to do the labs at a distance location. These methods have been found very useful for delivering the labs of the courses that do not require bulky and costly laboratory equipment such as industrial robots, trainers, test instruments, manufacturing machines etc.²

In order to deliver effective laboratory sessions, many institutions have adopted the use of computer software which can simulate the laboratory equipment including all of the required parameters so that their distance learning students could interact with them using a realistic virtual environment throughout their computers³. With the recent advances in computer graphics, the realistic visualization systems such as Virtual Reality (VR) has also been a popular enhancement to virtual laboratories as it improves the interaction with the subject studied. The VR technology, providing sense of reality and an impression of “being there”, has been increasingly employed in various training and educational applications in the areas of design and manufacturing, industrial automation, control and robotics^{4,5}.

The VR-based education has been reported to have several advantages such as: the flexibility that it gives to the students in performing assigned laboratory tasks remotely at a time which they feel comfortable; the ability to repeat experiments multiple times and learn by making mistakes; embedded explanations of theoretical concepts into the scenery where the laboratory experiment

takes place⁶. However, despite its advantages, the conventional VR technology, which is based on desktop software and specialized immersion equipment, is often considered as a high-end application in distance education. The higher education institutions find the conventional VR technology difficult to apply in distance learning for the following reasons: i) VR developers often rely on plugins or native applications and ask their users to download and install custom software in order to deliver a hardware-accelerated 3D experience. With the high complexity of the VR software, it is hard for average distance learning students to setup and operate the VR systems on their own without significant help, ii) cost of the VR software, computational hardware and the hardware required for special immersion effects is too high for students to afford. The cost of VR equipment is often found too high for the institutions to purchase them for large quantities for their distance learners. In addition, with conventional VR authoring tools, educators need advanced programming and modeling skills for developing custom VR sceneries for delivering their curriculum.

In this paper, we propose an approach for educators to develop a low-cost alternative to conventional VR platforms, suitable for use in distance engineering education. The proposed approach uses a web-based immersive 3D graphics visualization tool called the WebGL (Web-based Graphics Library), which is based on open-source modeling and programming tools, for creating 3D contents and interactive simulations to provide virtual laboratory education in engineering courses. The content generated through WebGL can work on any platform including desktop and mobile devices on in any compatible web browser without needing to install additional and costly software, app or plug-in. Hence, in order to interact with the virtual laboratory, the students can use a significantly less expensive interactive viewer such as Oculus Rift⁷ or Google's Cardboard VR, which is cut out of pieces of cardboard and folded into a 3D viewer for smartphones⁸.

This paper first describes the design methodology of WebGL-based virtual laboratory platforms suitable for distance engineering education. Second, it presents a prototype implementation of the proposed methodology in a senior-level distance learning course on industrial automation at Miami University, Department of Engineering Technology. We discuss the advantages and disadvantages of using WebGL as a basis for visualization in creating virtual laboratories for distance education, as well as predictions regarding further development of existing platforms and environments.

2. Overall Methodology: Building Virtual Laboratories with WebGL

WebGL is a cross-platform 3D graphics Application Programming Interface (API) that uses OpenGL⁹ as shading language to create 3D drawing context for web-based graphics visualization. It allows developers to use the 3D rendering hardware of their computers from within their web browser using standard JavaScript and Hypertext Markup Language (HTML)⁹. The WebGL standard is managed by the non-profit Khronos Group, who released its

specification 1.0 on March 3, 2011 ¹¹. The 3D graphics content can be cleanly combined with other web content that is layered on top or underneath the 3D content. This type of web-based graphics programming is ideally suited for creating dynamic, web-based 3D applications in the JavaScript programming language. The WebGL is currently supported by many leading web browsers including Apple (Safari), Google (Chrome), Mozilla (Firefox), and Opera (Opera). The WebGL technology has been increasingly used in various web-based simulation applications for scientific visualization, education, entertainment and gaming purposes ¹².

The approach presented in this paper uses a four-step methodology to build 3D, interactive simulation models to use as virtual laboratories for distance learning courses. The proposed methodology for building such simulation models with WebGL is shown in Figure 1. The overall process is based on creating a 3D model to replicate real, existing laboratory equipment in order to provide the distance learning student users a virtual experience for interacting with them using web. Hence, it is essential to identify the real physical equipment to-be-modeled at the beginning of the modeling process. At this point, it is also important to identify the learning objectives of the laboratory experience so that the level of detail of the simulation scene can be optimized accordingly. It is typically not feasible and cost-effective to incorporate all aspects of the laboratory environment in the simulation environment.

The modeling process with WebGL starts with creating a 'scene'. The scene represents the environment or the 'virtual world' which houses all the 3D content representing the laboratory equipment. A WebGL scene is physically implemented on the 'canvas' feature of HTML5, that can be used to draw graphs, create animations, or even do real-time video processing or rendering. The WebGL 'scene' involves mainly the three elements in order to be able to display and animate the 3D content in the canvas, namely: renderer, lights and cameras. The renderer displays the scenes in a realistic 3D forms and it helps sequencing the timing events i.e. animations. The lights help illuminating objects so that they can be viewed by the users. Finally the cameras allow viewing the objects in 3D environment in various angles. The users have the flexibility to add or remove as many and different light sources and cameras to the created scene. WebGL allows adding various dynamic camera types to enhance user interaction.

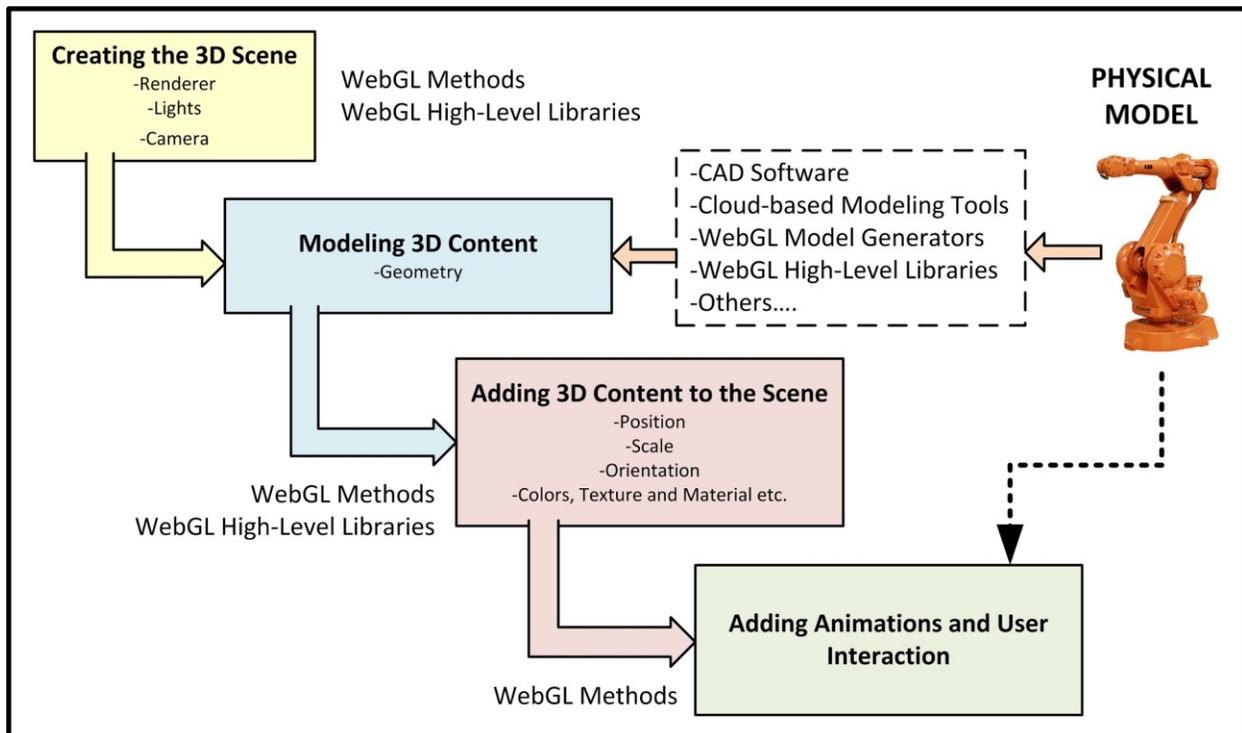


Figure 1: Overall Methodology of building VR simulation models with WebGL

In order to ease the modeling and development process with WebGL, several high-level JavaScript programming libraries have been introduced. Three.js, GLGE, SceneJS, PhiloGL are among the popular WebGL libraries. These libraries are intended to make it easier to author complex 3D computer animations that display in the Web-browser without use of any plug-in. For developing the proposed virtual laboratory in this paper, we mainly used Three.js as an authoring library. The Three.JS is a single JavaScript file, which can be included within a web page by linking to a local or remote copy using the following statement:

```
<script src="js/three.min.js"></script>
```

Three.js provides several library functions for creating a scene, adding cameras, lights and objects to the scene and setting the materials of the objects. The library also includes several features for easy generation of special effects and animations such as: Anaglyph, cross-eyed and parallax barrier, armatures, forward kinematics, inverse kinematics, morph and key frame. By using Three.js, 3D models can easily be embedded into the web browser, thus, providing a fluent web-based 3D interaction to the users ¹³.

The next critical step in building a simulation is the modeling the 3D representation of the real laboratory equipment. WebGL supports the use of various methods for modeling the 3D content. For example, the 3D content can be generated using 3D modeling CAD software such as AutoCAD™, SolidWorks™, Blender™ etc. or by using cloud-based 3D modeling tools (e.g.

Sketchup™, Sketchfab™, TinkerCAD™ etc.) and they can be imported into the scene. In addition to the third parts CAD platforms, one might use WebGL's native methods in JavaScript in order to generate 3D graphics objects. In WebGL, objects are built using sets of vertices, each of which has a position and a color; by default, all other pixels' colors (and all its other attributes, including position) are computed using linear interpolation, automatically creating smooth gradients ¹³.

Using a high-level WebGL graphics library such as Three.js might be useful as it provides library functions for generating various common objects such as: boxes, spheres, cylinders, cones etc. without extensive graphics programming skills. The object creating requires the definition of geometry. Once the 3D object geometries are created, or loaded to the scene from an external CAD file, the next step is to define how they should appear in the virtual environment. WebGL uses various library functions to allow users to set materials, color and texture of the objects. In addition, position, scale and orientation of the objects in the scene must also be set.

The JavaScript code given in Figure 2 demonstrates the process of creating a cylinder with 20 units in height, 4 units in radius using the Three.js library. As the first line describes the geometry (i.e. cylinder), the second line defines the material (i.e. colors, shininess, transparency etc.). Three.js uses hexagonal numbering system to describe colors. Once the object created is added to the scene with the **scene.add()** method, its coordinates, scale and orientation can be set by referring to its corresponding property.

```
var geometry = new THREE.CylinderGeometry( 4, 4, 20);
var material = new THREE.MeshBasicMaterial( { color: 0x00ff00 } );

cylinder1 = new THREE.Mesh( geometry, material );
scene.add( cylinder1 );

cylinder1.position.set( 0, - 10, 0 );
cylinder1.scale.set( 0.17, 0.12, 0.17 );
cylinder1.rotation.y=-1.57;
```

Figure 2: JavaScript code excerpt for creating a simple object in WebGL using Three.js

Another important step in the process of building virtual worlds is adding animations and interactive user controls to the scene. WebGL does not have built-in animation capability. However, it allows frame-based animation by means of rendering graphics and updating them at up to sixty frames per second. This feature provides several options for animating 3D content. Animations can be used to update the appearance and position of the objects in a WebGL scene. This includes: transformation of geometry, textures, materials, lights and cameras. In WebGL

environment, every 3D object that is added to the scene is assigned a name in JavaScript. Objects can move, rotate and scale, or follow paths; geometry can bend, twist and change into other shapes; textures can be moved, scaled, rotated, and scrolled, and have their pixels modified every frame.

It is important to note that, for realization of virtual laboratories that can simulate laboratory equipment and display educational concepts and theories to learners in action, the animations and user interaction play a highly essential role. Figure 3 shows a sample code taken from our implementation of the proposed methodology that is responsible for moving a bottle object on a conveyor belt in the virtual simulation environment. Axis x represents the direction that the bottle moves, while the parameter *position* is adjusted to place the bottle in an appropriate position.

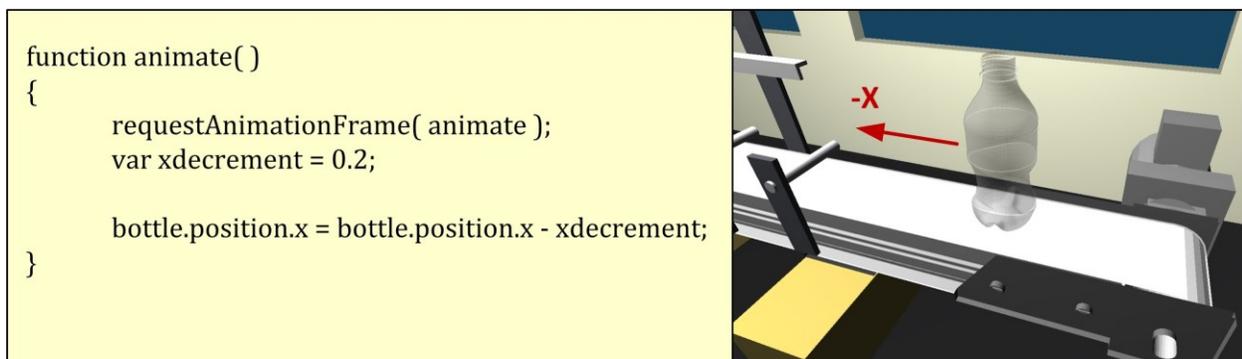


Figure 3: Sample code and simulation screenshot showing a basic animation of a bottle moving on a conveyor belt.

There are quite a few online resources including hands-on tutorials ^{14, Error! Reference source not found.} ^{15, 16}, which the educators may find them useful to learn more about WebGL animations and creating user interaction for creating dynamic virtual laboratories.

3. Prototype Implementation: Virtual Bottle Filling Process Trainer Unit

In order to measure the effectiveness of the proposed web-based simulation method in distance laboratory education, a pilot study has been performed in the ENT 402 – Industrial Automation Lab, a senior-level distance learning course in the Miami University, Department of Engineering Technology.

The Department of Engineering technology offers an ETAC/ABET accredited B.S. Electromechanical Engineering Technology program which has a distance learning component that connects with ten community colleges within a 300 mile radius of the main campus. The students from various geographic locations take courses from the Miami University campus remotely via the Interactive Video Distance Learning (IVDL) system.

One of the most challenging aspects of this program is the student laboratory experience. As per, required by the ETAC/ABET, in order for students to fully understand their operation, they need to work on the appropriate equipment for the laboratory component of each course of the program. Laboratory experiments for the program courses are designed for pedagogical purpose in order to meet the educational outcomes discussed by the ABET ¹⁹. For distance learning component, the program uses portable, mobile lab instruments which are shipped to the distance learning students for performing the laboratory experiments in the subject areas such as: data acquisition, signal processing, sensors, Programmable Logic Controllers (PLCs), hydraulics and pneumatics, and motor control ^{1,3}.

The ENT 402 Industrial Automation Lab is such a distance learning course of the program that is offered in both on-campus and distance learning formats, and it uses IVDL to connect to distance learners. The ENT 402 is predominantly a laboratory course, which covers a wide range of topics related to industrial automation systems including, industrial networking, Supervisory Control and Data Acquisition (SCADA), Human Machine Interfaces (HMI), remote control and monitoring of industrial processes. The course assumes that the students have a basic knowledge of PLCs and programming skills using ladder-logic. As a course requirement, students must purchase a lab kit that contains an educational PLC and peripheral accessories.

For delivering most of the most of the laboratory practices of this course, a bottle filling process trainer unit is utilized. The trainer consists of a set of PLC-controlled pneumatic stations in order to control and monitor the sequence of actions needed for filling capping and labeling 32ml coke bottles filled with various fluid flavors. Figure 4 presents the hardware of the bottle filling process trainer located in the Internet of Things and industrial Wireless Networking Lab (IoT-iWin Lab) of the Department of Engineering Technology at Miami University of Ohio. The trainer is composed of 4 different stages, namely: flavor selection, filling, capping and labeling. Two automated conveyor belts and a turntable are used to transport bottle(s) between the stages. The process starts with selection of a flavor from the three available flavors (Cola, Dew and Red). Once a flavor is selected, an empty coke bottle proceeds to the filling station and waits for the filling valve to be activated. The bottle is then filled with the selected flavor and it proceeds to the capping station. It then travels to on to the labeling and exits the system once it is labeled. The stages are deployed in a distributed fashion in order to demonstrate the flexibility and configurability, that is, each stage is controlled by a dedicated PLC.

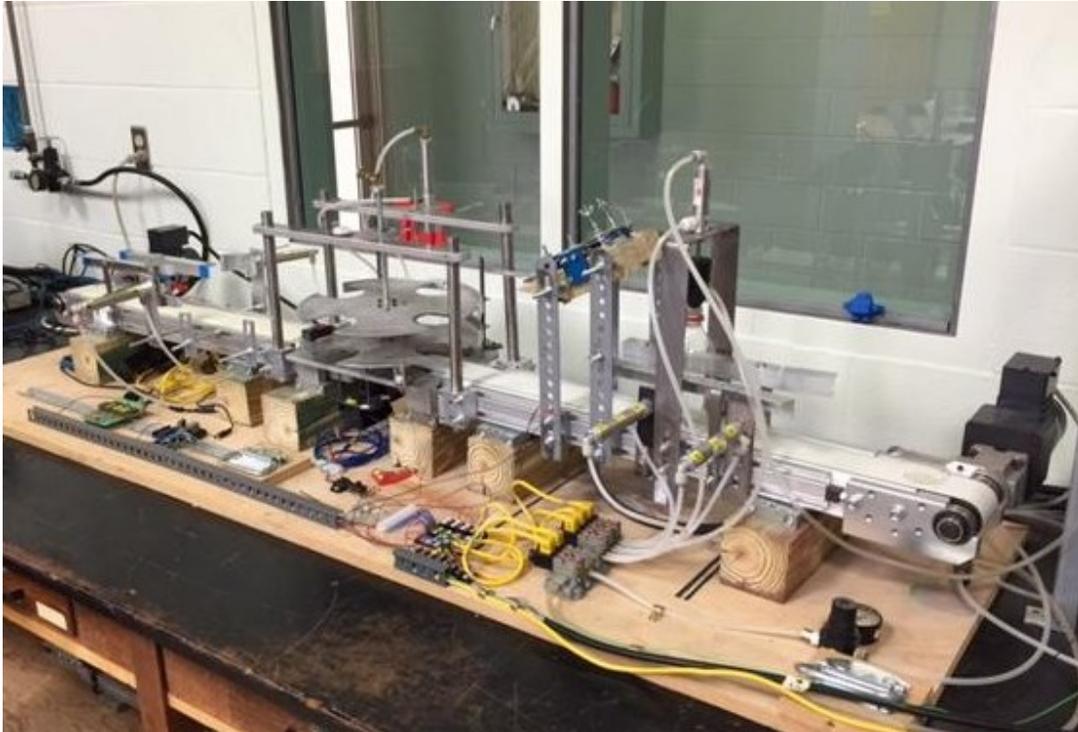


Figure 4: Bottle filling process trainer used in the ENT402-Industrial Automation Lab

In given lab assignments using this equipment, the students first need to program corresponding PLCs in order to perform the low-level control of the bottling process stages. In another assignment, the students create an HMI using Microsoft Visual Studio™ and have it communicated with the PLC through client-server networking for supervisory control and monitoring of the entire bottling process. For the final project of the course, students develop an SQL-based web server using HTML5 and have the users remotely monitor and control of the process control unit through web-client interfaces. Students are given basic instructions on HTML5, Cascading Style Sheets (CSS) and JavaScript coding in order to generate web user interfaces to control the bottling system from any platform including tablet computers and smartphones.

The course requires that all the lab assignments to be completed in the laboratory. Traditional, on-campus students conveniently utilize the trainer equipment in the physical laboratory space and complete their lab assignments. However, due to the size and cost of the bottle filling process trainer, it is not feasible for the university to make one available for each distance learning student. Hence, the distance learning students are not able to access the actual trainer in order to test their programs and complete their assignments.

This fact also constitutes the main motivation behind the proposed study. In order to provide the distance learners a fair access to the trainer and let them test their programs, a WebGL-based VR

simulation model has been developed using the methodology described in this paper. Figures 5 and 6 demonstrate snapshots from the prototype implementation of the virtual bottle filling station in a web browser.

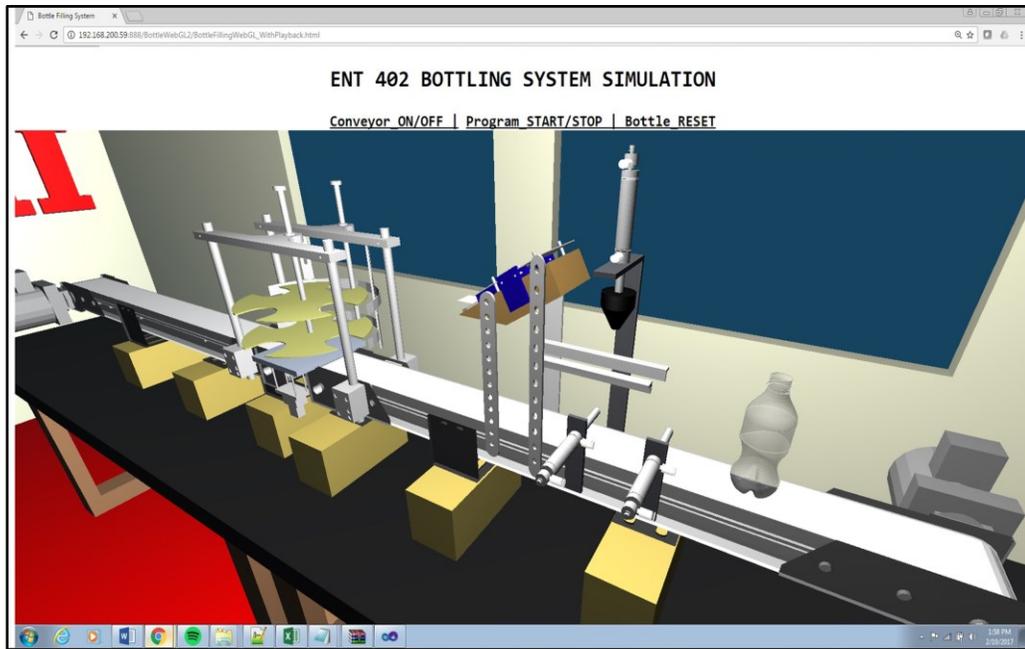


Figure 5. The implementation of virtual bottle filling trainer using WebGL and Three.js library

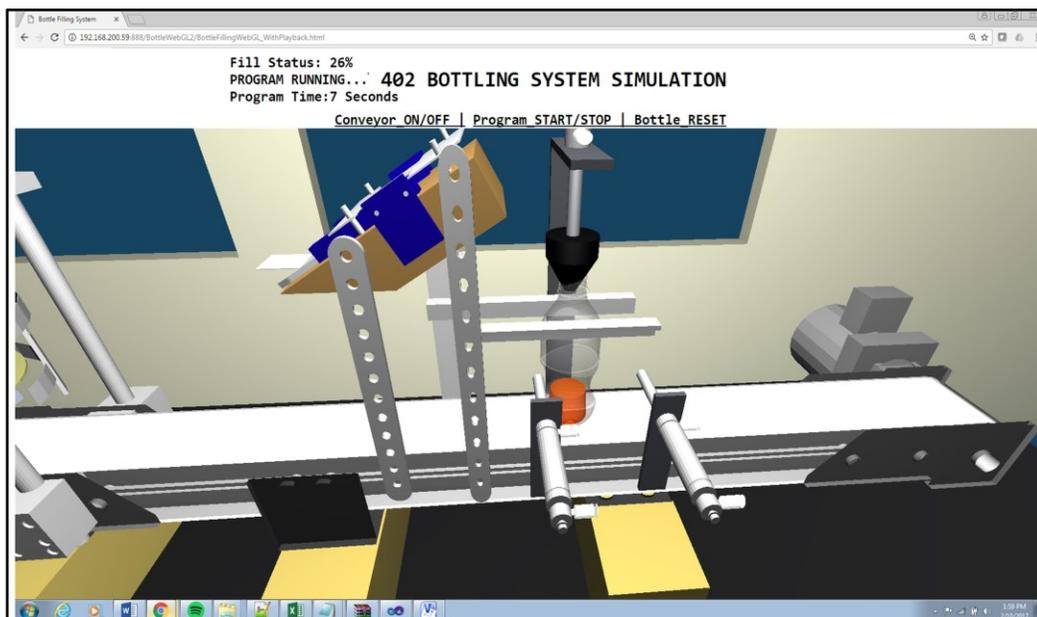


Figure 6: Filling stage on the virtual bottle filling trainer

For developing the bottle filling trainer virtual simulation model, all of its hardware components have been modelled based on their actual geometry, color, texture and material specifications in order to give distance learners a real feel of working with it. A combination of methods used for building the geometry of the components such as: AutoCAD™, TinkerCAD™ and Three.js methods using JavaScript. Each modeled system component has been programmed for performing controlled animations. For example, a valve is activated to fill the coke bottle with the liquid flavor selected by the students. Upon completing the filling process, a pneumatic cylinder lets the bottle proceed to the next stages for capping and labeling. The simulated WebGL model of the bottle filling trainer has been deployed on an HTML page, with an easy-to-use interface that allows students to view and control the virtual laboratory trainer in the VR environment.

The simulated system components have been programmed to respond the user actions. Presented virtual simulation model consists of two interaction modes: the playback mode, and the online mode. The playback mode consists of a ‘fixed system scenario’ which is hard-coded in the simulator. The ‘fixed scenario’ is an animated sequence of actions and is used to demonstrate the students the operation sequence of each stages. Before starting to program the trainer, the distance learning students are asked to view the operation of the system using the playback mode. By viewing the system in the playback mode, students get an idea on the “to-be” model of the system and develop their PLC code to control the stages of the system.

For better viewing the virtual model of the laboratory, the distance students are encouraged to use an immersive VR viewing device. Because digital see-through mixed reality glasses are expensive to use with distance learning classes, the Google Cardboard concept utilizing students’ smartphones are used to visualize the virtual laboratory model. Students need to purchase just a low cost Google Cardboard™, and a Bluetooth™ controller ²⁰ (See Figure 7).



Figure 7: Google Cardboard™ VR Viewer (left and center) ⁸, Wireless Bluetooth™ Game Controller (right) ²⁰

The online mode of the system allows all the actions of the virtual laboratory to be controlled by external interfaces. By using the online mode, the students can link their PLCs to the simulation model and control the simulated system actions by running their programs on their PLCs. This gives the students an opportunity to immediately test their programs for errors by observing its actions on the simulation model. Distance learning students use the proposed web-based simulation model in the playback and online modes to view the operation of the bottle filling trainer and develop their PLC programs. At the end of the laboratory session, the distance learning students upload their completed PLC codes on the Learning Management System. The instructor then downloads and runs each PLC program on the actual bottle filling trainer system in order to test its functionality. A web-controllable AXIS™ network camera is utilized at this stage for providing a visual feedback to distance students while their programs are run on the actual system. The web-camera is also equipped with a pan & tilt system. If desired, the distance students can remotely move the camera, pan, tilt and adjust their view angles as needed.

4. Evaluation of Usability and Learning Experience

In order to assess the distance learning students' experiences with the proposed virtual simulation system, we are currently implementing a feedback survey, which has been designed as a questionnaire to measure the proposed VR system's usability, and the students' general experience with the virtual laboratory.

For designing the feedback survey, we have utilized the five subscales of the SUMI usability assessment method ²¹. The subscales are defined in SUMI are namely; Efficiency: reflects the degree to which the software helps the user accomplish their task. Affect: measures the emotional response of the user to the software. Helpfulness: indicates whether system is relatively self-explanatory, and for which the help system and documentation are good or not. Control: shows the degree to which users feel in control of the software, rather than being controlled by the software. Learn ability: measures how quickly and easily the users felt they could master the software or a new feature of the software. According to these scales, a system that achieves a score in the range of 40-60 is comparable in usability to most of successful commercial software products. However, these subscales are defined for usability analysis for general software. Hence, in our survey, some additional application-related questions are particularly designed for the usability of the VR system using some of the heuristics as defined by Sutcliffe *et.al.* in 2004 ²².

We are currently in a process of collecting data with the proposed system prototype. We will publish the results of the surveys in future publications.

5. Conclusion and Future Works

An approach for developing web-based virtual laboratories for distance engineering education has been presented in this paper. The proposed approach uses WebGL standard on compatible web browsers to provide users with interactive, VR environments for experimentation of the laboratory concepts taught in engineering courses.

In order to demonstrate the implementation of the presented method, we have created a VR model of a bottle filling trainer unit that is used in a senior-level distance learning course on industrial automation. The functioning prototype of the virtual bottle filling trainer has been available for distance students online via a web browser. Using the virtual trainer, the distance learning students can perform virtual experiments with fluent 3D interaction realized by the latest HTML5. Students also use web cameras to connect the actual laboratory remotely in order to validate the experiment results with a strong sense of presence.

According to our experience with the prototype virtual laboratory, the web-based VR system gives students a sense of “being there,” so that the students can feel free to develop solutions to “what if?” scenarios in real time with 3D objects of real-time laboratory equipment. Hands-on experience through virtual laboratory gives direct experience to students thereby increasing their capability to apply what they have learnt. The use of computer graphics to render simulations accurately and effectively in 3D can help students to directly translate theory into practice. The virtual laboratory environment has also given the instructors the opportunity to introduce distance students the complex and expensive laboratory equipment in a cost-effective way. In addition, due to its web-based nature, students many students have found the presented prototype system easy-to-use. The WebGL technology used as a basis for modeling the virtual laboratories is a convenient tool as it provides an inexpensive alternative for sophisticated and costly immersive VR simulators. Overall, the web-based VR promises to be a great potential in distance education as it engages distance learning students to the theories by providing them an easy-to-use VR platform to perform hands-on lab experiences.

WebGL-based virtual laboratory prototype presented in this paper is currently under development for various improvements. The VR platform currently requires more research and development in creation of content which accurately depicts theory and that which can give the right perception to students using it. We are presently working on graphical enhancements of the virtual laboratory model in order to make it look more realistic and to provide a better virtual learning experience. In addition, we are still developing the ‘online mode’ operation in order to provide students a full access to control the actions in the virtual laboratory. We anticipate publishing the further developments on the virtual laboratory platform along with the usability survey results in the near future.

References:

1. Bal, M. (2012), Virtual Manufacturing Laboratory Experiences for Distance Learning Courses in Engineering Technology, *Paper presented at 2012 ASEE Annual Conference & Exposition, San Antonio, Texas.* <https://peer.asee.org/22218>
2. Hashemipour, M., Manesh, H.F., Bal, M. (2011), A modular virtual reality system for engineering laboratory education, *Computers Applications in Engineering Education (John Wiley & Sons)*, 19 (2011) 305-314.
3. Bal, M. (2014), Assessment of Remote Laboratory Practices in Engineering Technology Distance Education, *Paper presented at 2014 ASEE International Forum, Indianapolis, Indiana.* <https://peer.asee.org/17168>
4. Bal, M., Manesh, H. F., Hashemipour, M. (2008), Virtual-reality-based information requirements analysis tool for CIM system implementation: a case study in die-casting industry, *International Journal of Computer Integrated Manufacturing*, 3(21), 2008, pp: 231-244.
5. Bal, M and Hashemipour, M. (2009), Virtual factory approach for implementation of holonic control in industrial applications: A case study in die-casting industry, *Robotics and Computer-Integrated Manufacturing*, 3(25), 2009, pp:570-581.
6. Burdea, G. C., & Coiffet, P. (2003). Virtual reality technology (Vol. 1). *John Wiley & Sons.*
7. Desai, P. R., Desai, P. N., Ajmera, K. D., & Mehta, K. (2014). A review paper on oculus rift-a virtual reality headset. *arXiv preprint arXiv:1408.1173.*
8. Google VR Cardboard. (2017). Available online: <https://vr.google.com/cardboard/>
9. OpenGL - The Industry's Foundation for High Performance Graphics. Available online: <https://www.opengl.org/>
10. Parisi, T. (2017). Programming 3D Applications with HTML5 and WebGL (E-Book), Chapter 2. WebGL: Real-time 3D Rendering, *O'Reilly Media*. Available online: <http://chimera.labs.oreilly.com/books/1234000000802/ch02.html>
11. WebGL - OpenGL ES 2.0 for the Web (2011). Available online: <http://www.khronos.org/>.
12. Lei, Z., Hu, W., & Zhou, H. (2016). Deploying web-based control laboratory using HTML5. *In Remote Engineering and Virtual Instrumentation (REV), 2016 13th International Conference on* (pp. 69-73). IEEE
13. Three JS-JavaScript 3D Library (2017). Available online: <https://threejs.org/>
14. Using shaders to apply color in WebGL, *Mozilla Developer Network*. Available online: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Using_shaders_to_apply_color_in WebGL
15. Parisi, T. (2017). Programming 3D Applications with HTML5 and WebGL, Chapter 5. 3D Animation, *O'Reilly Media*. Available online: http://chimera.labs.oreilly.com/books/1234000000802/ch05.html#animating_characters_with_skinning
16. Sopylo, M. (2013), WebGL with Three.js: Models and Animation, *envatotuts+*, Available Online: <https://code.tutsplus.com/tutorials/webgl-with-threejs-models-and-animation--net-35993>.
17. Dirksen, J. Chapter 9. Animations and Moving the Camera, *Learning Three.js - the JavaScript 3D Library for WebGL*, Packt Publishing Limited. Available Online: <https://www.packtpub.com/mapt/book/web-development/9781784392215/9>
18. Pettit, N. (2013), The Beginner's Guide to three.js, *treehouse*. Available Online: <http://blog.teamtreehouse.com/the-beginners-guide-to-three-js>
19. ABET – Accreditation Board for Engineering and Technology. <http://www.abet.org/>
20. High Quality Wireless Bluetooth Game Controller Gamepad Joystick for Android iOS Smart Phone for iphone for sumsung (ET261). Available online: <http://smartphoneswag.com/?product=100-new-arrival-wireless-bluetooth-game-controller-gamepad-joystick-for-android-ios-smart-phone-smartphone-hot-sale-et261>

21. Kirakowski, J., & Corbett, M. (1993). SUMI: The software usability measurement inventory. *British journal of educational technology*, 24(3), 210-212.
22. Sutcliffe, A., & Gault, B. (2004). Heuristic evaluation of virtual reality applications. *Interacting with computers*, 16(4), 831-849.
23. Congote, J., Segura, A., Kabongo, L., Moreno, A., Posada, J., & Ruiz, O. (2011, June). Interactive visualization of volumetric data with WebGL in real-time. In *Proceedings of the 16th International Conference on 3D Web Technology* (pp. 137-146). ACM.