Plug-in Free Web Based 3-D Interactive Laboratory for Control Engineering Education

Wenshan Hu, Zhongcheng Lei, Hong Zhou, Guo-Ping Liu, *Fellow Member, IEEE*, Qijun Deng, Dongguo Zhou and Zhi-Wei Liu, *Member IEEE*

***Abstract*—This paper introduces the design and implementation of a plug-in free online 3-D interactive laboratory based on NCSLab framework. The system relying only on HTML5 provides full supports for control engineering experimentation. The users are allowed to design their own control algorithms and apply them to the remote test rigs. Using the web-based interface, multiple widgets such as real-time charts, virtual gauges and live images are available to customize the monitoring interfaces. To enhance the sense of immersion, 3-D animations which are synchronized with the remote experimental processes are also provided. The users can watch and interact with the remote experiments through the 3-D replicas. Various HTML5 based toolkits are integrated seamlessly under the NCSLab framework. NCSLab provides visualized services for the whole process of control experimentation including remote monitoring, tuning, configuration and control algorithm implementation. As the network delay could disturb the 3-D representation, a communication scheme using web protocols is also implemented. The feedback from teaching shows the general acceptance and effectiveness of NCSLab is notably high. As most existing online laboratories adopt either native applications or plug-ins, the methodologies and technologies used in NCSLab could be insightful for other online laboratories towards web-based cross-platform systems.**

***Index Terms*— Control engineering education, laboratories, online services, remote monitoring, virtual reality.**

I. Introduction

EXPERIMENTATION which relates the theories and concepts to the real world observation plays the same important role as lectures in the engineering education. However, many educational institutions may not be able to afford all the costly equipment and related maintenance costs. To address the problem, online laboratories which allow the users to access laboratory resources and carry out experiments through internet have appeared. Online laboratories include the remote laboratories [1]-[4] which use practical equipment, virtual laboratories [5], [6] which use simulated equipment and hybrid laboratories [7], [8] which support both of them. As experimentation becomes online services, laboratory resources are shared among educational institutions [9], [10].

Encouraged by the advantages of the online experimentation, many online experimental systems have been established worldwide in the last decades. These systems almost cover all the areas of engineering education, including control engineering [11]–[13], electric and electronic engineering [14]–[17], mechanical engineering [18], industrial electronics [19], [20] and biomedical engineering [21], [22].

Currently the client sides of most online laboratories are deployed with either native applications [23], [24] or web-based solutions with requirement of plug-ins such as Easy Java Simulations (EJS) Applets [11], [25], [26], Adobe Flash [27], LabVIEW run-time engine [12], [28] or other specialized third-party software [16], [29]. The native applications are flexible, powerful. It is easy to develop experimental systems with rich graphical interactivities. However, the drawback is the universality as they are always bound to certain hardware or software architectures.

The web applications offer the possibility of cross-platform accessibility with more portability and less intrusiveness [30]. Theoretically, web applications are compatible with all the existing and even future terminal devices with web browsers. However, due to the limitation of old HTML (HyperText Markup Language) framework, traditional web-based solutions have less competitive advantages over native applications. Visual components such as real-time charts, virtual gauges and 3-D animation cannot be embedded into the web interfaces without the help of plug-ins. As a compromise, plug-ins which enhance the power of web applications could also cause unexpected compatibility and security problems.

Despite the limitation and difficulties, building fully web-based, plug-in free cross-platform and powerful online laboratory architectures with rich graphically interactive features has been a major trend for years [2], [30]-[32]. It has only been recently possible as HTML5 standard was finalized in late 2014 and most of the mainstream web browsers started to provide full support for the new standard. HTML5 enhances

Manuscript received May 17, 2016; revised September 13, 2016; accepted November 21, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61374064, Grant 61333003 and Grant 61673303.

W. Hu, Z. Lei, H. Zhou, Q. Deng, D. Zhou and Z. -W. Liu are with the Department of Automation, School of Power and Mechanical Engineering, Wuhan University, Wuhan 430072, China. (phone: +86-27-68772267; e-mail: wenshan.hu@whu.edu.cn).

G. -P. Liu is with the School of Engineering, University of South Wales, Pontypridd CF37 1DL, UK and is also with the CTGT Center, Harbin Institute of Technology, Harbin 150001, China.

the support for building more powerful web applications previously only possible for native applications and plug-ins.

There are already attempts for HTML5 based online laboratory recently. In the area of electronic education, remote panels in LabVIEW for a BJT (Bipolar Junction Transistor) amplifier are converted to HTML5 [33]. Labicom [34] is a commercial architecture which provides APIs for online laboratories to be built on top of its infrastructure. GOLDi [35] provides fixed 2-D graphical interfaces using HTML5. The 3-D visualization has not been implemented so far. The client side of the system introduced in [36] is realized based on web technologies. The source codes were initially designed in Java and then compiled into JavaScript using Google Web Toolkit. WebLab-FPGA-WaterTank [31] discusses the possibility of a universal hybrid model. The new idea is verified by an example of a 3-D virtual water tank controlled by a real FPGA controller. The client side of iSES Remote Lab SDK [37] is also implemented using web technologies. However, iSES concentrates on the modular hardware architecture for the quick deployment. For the software part, there is little particular information on the 2-D widgets for the developers and 3-D is not yet supported.

All of these literatures claim the supports for web technologies HTML5, but lack of information regarding the systematic implementation. As HTML5 is still a relatively new concept, there is no full picture on how a fully functioning cross-platform online laboratory is developed using HTML5 so far. In this paper, the detailed design and implementation for NCSLab (Networked Control System Laboratory) framework [38]-[40] are introduced. NCSLab is a complex plug-in free online 3-D control laboratory relying only on HTML5. It provides full supports for the whole process of control engineering experimentation including remote monitoring, tuning, measurement, visual configuration and control algorithm implementation.

Based on HTML5, NCSLab has brought in various interactive features including 3-D interactivity, virtual gauges, real-time charts, and client side camera. To enhance the sense of presence, 3-D replicas of the remote control processes are synchronized with the experimental processes so that users are able to “monitor” and even “interact” with the remote experiments.

In order to achieve complex functionalities with rich 2-D and 3-D graphical interactive features, NCSLab adopts modular software architecture. Many HTML5 based technologies are integrated together seamlessly. The real-time communication scheme based on web protocols is also designed to maintain the synchronization between the web interfaces and remote experimental processes with minimum latency.

II. Features and Architecture of Web Based 3-D Laboratory - NCSLab

NCSLab was first established in University of Glamorgan (now known as University of South Wales) in 2006. It adopted the web technologies from the very beginning. One of the main objectives of NCSLab is to build a service-oriented, plug-in free and cross-platform laboratory which offers complex online experimental services for university students covering every aspect of control engineering experimentation with enhanced graphical interactivities and sense of presence.

TABLE I
Evolvement of NCSLab Framework

Year	Features	Toolkits	Plug-ins
2007 [38]	Fixed monitoring interface with charts and real-time videos	Java Applet, iframe	Java Applet
2009 [39]	User customized algorithms, visual configurable monitoring interface, virtual gauges, charts, real-time videos	Flash, Ajax, Yahoo UI, Dojo	Flash
2012 [40]	3D animation synchronized with remote experiments was added	Flash, Away 3D, Stage 3D, Yahoo UI, Dojo, FusionCharts (Flash version)	Flash
2016	Interactive 3D animation and web side camera were added. All plug-ins were eliminated by using HTML5	HTML5, WebGL, Three.js, Yahoo UI, FusionCharts (HTML5), Canvas.js	None

During the ten-year intensive development, it has evolved through four major versions as summarized in Table I. New features and new supporting technologies were gradually added into the NCSLab framework which made it more powerful and reliable. However, the objective towards plug-in free has only been recently possible with the help of HTML5.

A. Current Architecture of NCSLab

Fig. 1 shows the architecture of the deployment of NCSLab in Wuhan University. It has a multi-tier structure which allows the test rigs distributed in different places to be integrated and accessed through a single web address. Users are allowed to carry out online experiments anytime anywhere using their web browsers.

As a hybrid online laboratory framework, both remote and virtual experiments can be integrated into NCSLab simultaneously. If the remote controller is connected to the physical test rigs through the hardware interface such as A/D and D/A converters, it is in the remote experiment mode. If the practical devices are replaced by the mathematical models, the controller actually “control” the mathematical representation of the practical test rigs. In that case, virtual experimental devices are created to simulate the behaviors of practical ones.

B. Customized Control Algorithms Using MATLAB/Simulink RTW

All the control algorithms are generated from Simulink block diagrams using MATLAB/Simulink RTW (Real-Time Workshop). The block diagrams designed in Simulink are converted into C codes automatically and compiled into executable programs using C compilers. Using the web interface provided by NCSLab, these executable programs are uploaded to the NCSLab server database.

For each test rig, NCSLab allows the teachers to setup several default control algorithms which are shared with everyone. These algorithms can be some classic control schemes such as PID and LQG. The users can download these algorithms to the remote controller and perform the experiments.

For advanced users, they are also encouraged to design their own control algorithms. They can download the templates of default control algorithms in the form of Simulink block

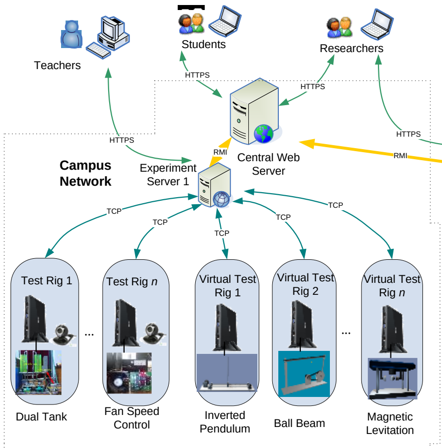


Fig. 1. Current NCSLab architecture in Wuhan University

diagram. Based on these default control algorithms provided by the teachers, they can also create their own control schemes.

C. Interactive Features of NCSLab

The web-based remote experimentation interface of NCSLab provides multiple widgets (charts, virtual gauges, sliders *etc.*) for the users to customize their user interface as shown in Fig. 2. The users are allowed to drag the widgets from the toolbar and place them into the workplace. These widgets can be relocated and resized freely. The internal signals and parameters of the RTW generated algorithms are normally organized as a tree structure as shown in Fig. 2. They can be selected to establish the interconnections with the widgets placed in the workplace.

The real-time interaction with the remote experimental process is also an important feature of NCSLab. Users are able to monitor the remote experiments, tune the parameters in real time through the widgets. Fig. 3 is a typical web-based experimental interface for a dual tank test rig. The PI control parameters can be changed by typing the new values into the textboxes. By dragging the virtual slider using the mouse, the water level set point for the remote control algorithms can also be updated automatically through the internet datalink.

D. 3-D Interactivities of NCSLab

In remote experiments, web cameras are deployed in front of the test rigs. The real-time images of the experimental processes can be transferred to the users through internet. However, this is no longer the case for virtual experiments. Without observing the practical experimental process, it could be difficult for the users to get the basic concepts of the experiments, especially for the beginners.

To address this issue, the 3-D replicas of the test rigs are rendered in the users' web based interface instead of the real-time videos. Through the datalink between the user interface and remote experimental processes, the 3-D models on the client side are synchronized with the internal signals of the remote virtual experiments. By monitoring the real-time animation on the web-based interface, the users are able to "observe" how the experiments are going on similar as they watch the live videos in remote experiments.

Moreover, the users are also allowed to interact with the virtual 3-D replicas. For example, they can change the valve

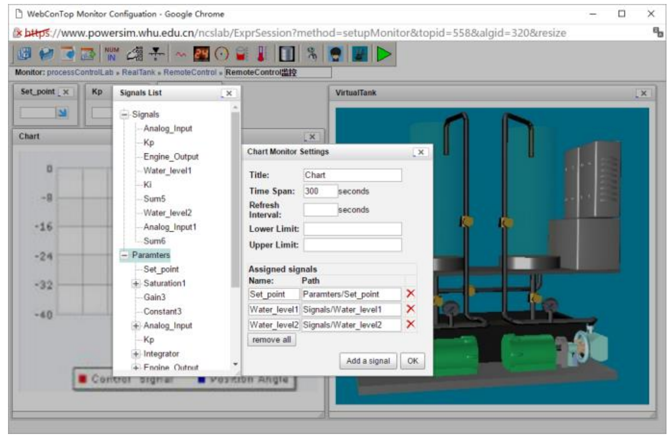


Fig. 2. Visual configuration interface

positions, set point cursors *etc.*, the corresponding parameters would also be updated through the network automatically.

Even for the remote experimentation with practical test rigs, the 3-D animations can also benefit the users. The web cameras can only provide the images from a fixed angle, but the 3-D models can be rendered and displayed from any perspective as the users are able to rotate, zoom in/out freely.

III. Design of Plug-in Free Web Based 3-D Laboratory

Plug-ins could induce security and compatibility problems. However, the use of them was inevitable for the early version of NCSLab and some other online platforms. The first version of NCSLab required users to install Java Applet, which was switched to Flash in 2009. However, plug-ins cannot be eliminated at that time due to the limitation of HTML4 framework.

The finalization of HTML5 standard in late 2014 had changed the situation. As an important part of HTML5, WebGL (Web Graphics Library) provides strong support for both 2-D and 3-D computer graphics. Some interactive features which can only be achieved by plug-ins previously are able to be realized directly using JavaScript, which makes it possible to build a plug-in free online laboratory.

A. Structure of Plug-in Free Web Interface for Remote Experiments

In order to achieve plug-in free web interface, various supporting toolkits (such as Three.js, Fusioncharts and Canvas.js) have been integrated together. A modular software architecture is created as illustrated in Fig. 4. Each module serves a certain function and is replaceable in case more advanced or convenient frameworks emerge in the future.

The kernel part of the web based interface is Yahoo UI (YUI), which is a free, open source JavaScript and CSS library for building richly interactive web applications. Using YUI, the developers can easily create dynamical DOM (Document Object Model) objects which are draggable and resizable inside the web pages.

The main task of YUI is to manage the various widgets such as 3-D animation, charts and virtual gauges in NCSLab. YUI creates a DOM object called Widget Container in which all the widgets are laid out. Each widget is also realized inside a

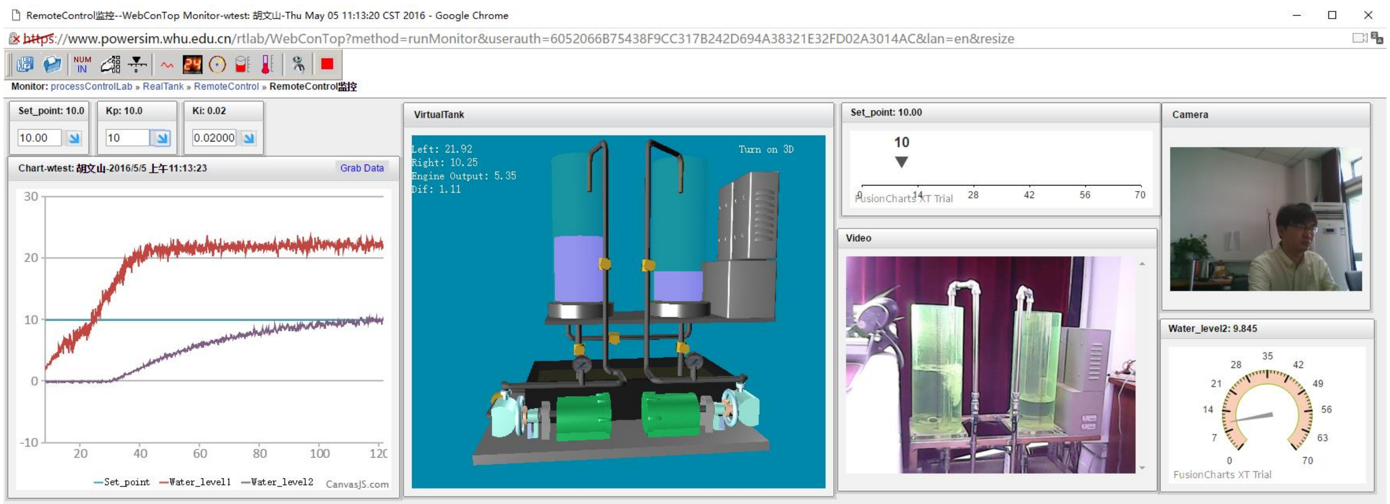


Fig. 3. Web-based experimental interface for dual tank control system

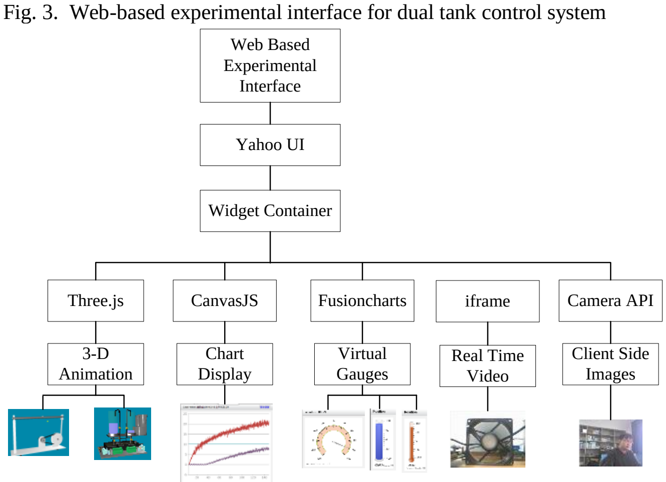


Fig. 4. Structure of web interface for remote experiments.

dedicated DOM object created by YUI. The users can create, move and resize the DOM objects freely, and the contents inside of the DOM objects are also changed correspondingly.

Under the YUI framework, various JavaScript toolkits are integrated as follow. 1) Three.js is adopted to render the 3-D animations which are synchronized with the remote test rigs. 2) CanvasJS is used to create dynamic charts to display the real time data. 3) Fusioncharts are integrated to build virtual gauges. 4) The real time videos are embedded inside iframe makeups which displays the images collected from separate video servers. 5) Moreover, the client side images are also collected using Camera API. The images of the users are also sent back to the central server. Therefore, the teachers are able to "see" who are actually conducting the experiments.

B. Configuration of Monitoring Interface

As a feature of NCSLab, the users are allowed to customize their own monitoring interface. Fig. 5 shows an example of the customized interface. There is a toolbar with all the widgets listed on the top. The users can drag them to the workplace freely to build the customized monitoring interface.

YUI library supports resizable and movable Panel object. Each widget is assigned a Panel rendered inside it. The users can resize and change the positions of the Panels. The layout of the customized interface can be setup.

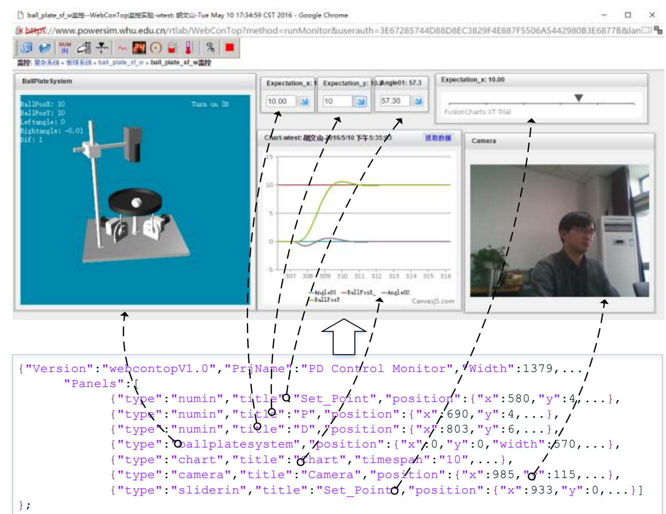


Fig. 5. JSON files for web based monitoring interface.

All the widgets are connected to the internal signals and parameters of the remote control algorithms. To build the connections, the users can double click the widget Panel. A pop-up overlay with a full tree structure listing all the signals and parameters will appear. The users can click the mouse on the tree structure to select the signal/parameters they want to monitor. Then the interconnection between the given widget and selected signal/parameters is established.

After the users finish the configuration of monitoring interface, the actual configuration data are organized in JSON format. They are transferred to the server and stored into the database.

C. Construction of Monitoring Interface

Fig. 5 shows an example of the monitoring interface during the experimental process. It is constructed according to the JSON files which are retrieved from the server database.

For each kind of widget, there is a corresponding JavaScript class designed for them. For example, the 3-D widget for the ball and plate test rig is called "ballplatesystem". As shown in Fig. 5, all the widgets used in the customized monitoring interface are listed in JSON format including their types, positions and related signal/parameters. By parsing the JSON configuration, all the widgets are built inside the workspace one by one and the web based monitoring interface is established.

As shown in Fig. 6, all the 2-D and 3-D widgets are laid out inside the Widget Container. For each widget, there is a YUI Panel created and placed at the specified position. No matter the types of the widgets, they can always be embedded inside web interface by rendering themselves inside the dedicated Panels. For example, the following code renders an angular gauge inside a YUI Panel:

gauge.render(GaugeDom);

in which *gauge* is the JavaScript object for the angular gauge and *GaugeDom* is the DOM object representing the main body of the corresponding YUI Panel.

IV. Plug-in Free Web Based 3-D Visualization

Apart from the real time video, widgets for 3-D visualization are also provided in NCSLab. The 3-D models of the experimental test rigs are rendered inside the monitoring interface. It is synchronized with remote experimental process through internet datalink.

As a major feature of HTML5 standard, WebGL is a JavaScript API which supports the rendering of interactive 3-D computer graphics within web browsers directly. It requires no plug-ins and allows the GPU accelerated 3-D rendering, which makes it possible for 3-D applications inside the web browsers.

Based on WebGL, there are many HTML5 3-D engines available in the market such as Unity, Unreal, Three.js and Babylon.js. Among them, Three.js which is a popular open source engine is adopted by the NCSLab.

A. 3-D Modeling

All the 3-D models are constructed according to the real dimensions. There are many commercial software tools available for the 3-D model design, such as 3ds Max, Solid Works and Pro/E. In this paper, 3ds Max is selected as the 3-D model development tool. The models designed in the 3ds Max are exported into the Wavefront OBJ format which can be parsed by Three.js. Fig. 7 is an example which shows the 3-D model of a ball and plate test rig being designed in 3ds Max environment.

3-D models with a lot of details always result in big size of the target files, which could cost long time for downloading these files in internet environment. Therefore, it is necessary to carefully balance the quality and complexity of the 3-D models. Normally, the size of the target files should not exceed 1M bytes.

B. Rendering inside the Web Based Interface

The 3-D models are rendered inside the DOM objects created by YUI. As shown in Fig. 6, each widget has a dedicated Panel in which there is a DOM object for the main body created by the YUI toolkit. Using the following codes, a renderer created for the 3-D animation is inserted into the corresponding YUI Panel:

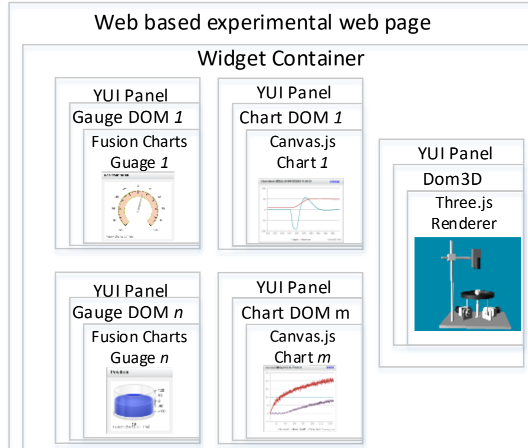


Fig. 6. Structure of web based monitoring interface

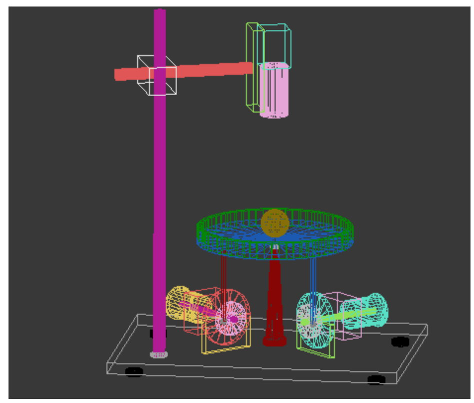


Fig. 7. Model of ball and plate system designed in 3ds Max

var renderer = new THREE.WebGLRenderer();
…
Dom3D.appendChild(renderer.domElement);

in which *renderer* is created for rendering the 3-D models and *Dom3D* represents the DOM object for 3-D widget.

V. Data Communication and 3-D Animation

To build a web based 3-D interactive laboratory, reliable data communication schemes among the client side, server and remote controllers are of great importance. In order to enhance the security, HTTPS which provides encrypted Secure Sockets Layer (SSL) is implemented instead of HTTP.

A. Data Communication Schemes

The internet communication for NCSLab is bidirectional. On one hand, widgets (both 3-D replicas and 2-D modules) in the web browser are synchronized with the data collected from remote experimental processes. One the other hand, the users' instructions (tuning parameters and changing algorithms *etc.*) need to be transmitted to the remote controller. From the communication point of view, three kinds of communication schemes are required to deal with the widgets in NCSLab.

1) *Scheme for 2-D widgets.* The examples of 2-D widgets are virtual angular gauges, sliders and thermometers *etc*. The status of 2-D widgets is updated every second. Therefore, they only need to request data using a timer and the continuous data stream is not necessary. However, the bidirectional communication is required as the user may trigger instructions through these widgets any time.

2) *Scheme for charts.* The chart modules display the trend of specified signals in real time. Therefore, a unidirectional data stream which retrieves all the continuous data without any interruption is necessary.
3) *Scheme for 3-D models.* The 3-D models are synchronized with the remote experimental processes according to the data from the datalinks. In order to construct the real-time animation without any distortion, a continuous data stream without data loss is required. As the users are allowed to interact with the remote experiments through the 3-D virtual models (such as the change of a valve position and the turn on/off of a switch), the instructions trigged by the users' interactions should also be sent to the remote controller through the datalinks.

B. Data Communication Channels

Fig. 8 illustrates the data communication channels for the multi-tier NCSLab architecture. For each test rig, there is a corresponding Experiment Engine deployed in one of the Tomcat Experiment Server. The Experiment Execution Module is the interface to the test rig. When a control algorithm is downloaded and executed in the controller, a TCP communication is established automatically between the Experiment Execution Module and the SCADA Server running inside the controller. The Experiment Execution Module collects the real-time data from the controller and temporarily stores them into a Real-Time Data Pool.

When a user starts to monitor the experimental process, the web page for the experimental interface is loaded by the web browser. After the loading process, a HTTPS communication between the widgets in the web interface and the corresponding Data Exchange Servlet is created automatically. On one hand, the Servlet transmits the latest data from the Real Time Data Pool to the client side. On the other hand, it also processes the users' instructions received from the HTTPS channels.

On the client side, the HTML web pages are rendered and the JavaScript codes are executed. The widgets are constructed and placed according to the user's configuration organized in JSON format. These widgets are synchronized with the real-time data collected from the remote experimental process through HTTPS connections.

To cope with the three kinds of widgets, two kinds of communication objects in JavaScript are designed. All the widgets exchange the real-time data with the server side through the two objects as the block diagram shown in Fig. 8.

1) *DataExchanger:* It is a bidirectional communication object. It requests the latest data from the server and updates the status of the 2-D widgets. If the user interacts with a certain widget, their instructions are also transmitted to the server side through the *DataExchanger*.
2) *DataStream:* The *DataStream* also requests the data from the server every half second. Different from the *DataExchanger*, it retrieves all the data generated after the previous request. Even though the data transmission is discrete, the fractional data are assembled to form a continuous data stream without any loss. For each chart or 3-D model, there is a dedicated *DataStream* object created. They retrieve and assemble the data into buffers, so the continuous data are always available for widgets.

C. Data Buffering for 3-D Visualization

The multi-tier communication channels can guarantee the reliable data exchange among the client side, server and remote experimental process. However, in order to achieve a vivid 3-D synchronization without any distortion, there are two issues to be considered.

1) *Network delay:* As the real-time data are transferred through the three-tier network structure, the network delay is inevitable and prominent, especially in the internet environment. The network delay could be random and unpredictable, which distorts the 3-D replay greatly.

2) *Packed data:* The advantage of the network communication is that a packet of data can be transmitted simultaneously. Normally, the internet communication prefers relatively large packets as they are more efficient. Frequent communication with small packets of data may increase the traffic load and result in a bigger time delay.

In order to get a vivid 3-D replay, minimum 25 FPS (Frames Per Second) must be guaranteed, which means the interval between two frames should be no more than 40ms. However, normally the *DataStream* object retrieves data every half second. Therefore, there is a contradiction between the rates of communication and 3-D replay.

To cope with the problem, a proper buffering algorithm is designed as shown in Fig. 9. All the real-time data are packed with timestamps. The timestamps and data are transmitted together to the client side browsers via the Experiment Server.

To properly buffer the data, there are two timelines adopted on the client side. One is the remote timeline which represents the timestamps packed with the real time data. The other is the local timeline which represents the internal client side time retrieved from the JavaScript *getTime()* method.

The two timelines are artificially misaligned for a small margin (normally one second) at the initialization. The local timeline always slightly lags behind the remote one, which guarantees the data required by the 3-D widgets for "current time" are always available in the buffer. The 3-D replay won't be interrupted by the network delay and jitter.

D. 3-D Animation Scheme with Buffering

All the latest data are properly buffered inside the *DataStream* objects. The 3-D animation widgets only need to concentrate on the motion control of the virtual models. A render loop under Three.js toolkit is created to update the 3-D scene up to 60 times per second. A callback function called *Animate()* is defined to manipulate the animations. There are various motion control APIs provided by Three.js toolkit. Using these resources, the 3-D models can be easily moved, rotated and scaled by the Motion Control Modules of the 3-D widgets.

For example, the 3-D animation for the ball and beam system is achieved by the following way. When the 3-D widget initiates, the models of the base, beam, lever arm and ball are loaded separately and placed at the initial positions. During the

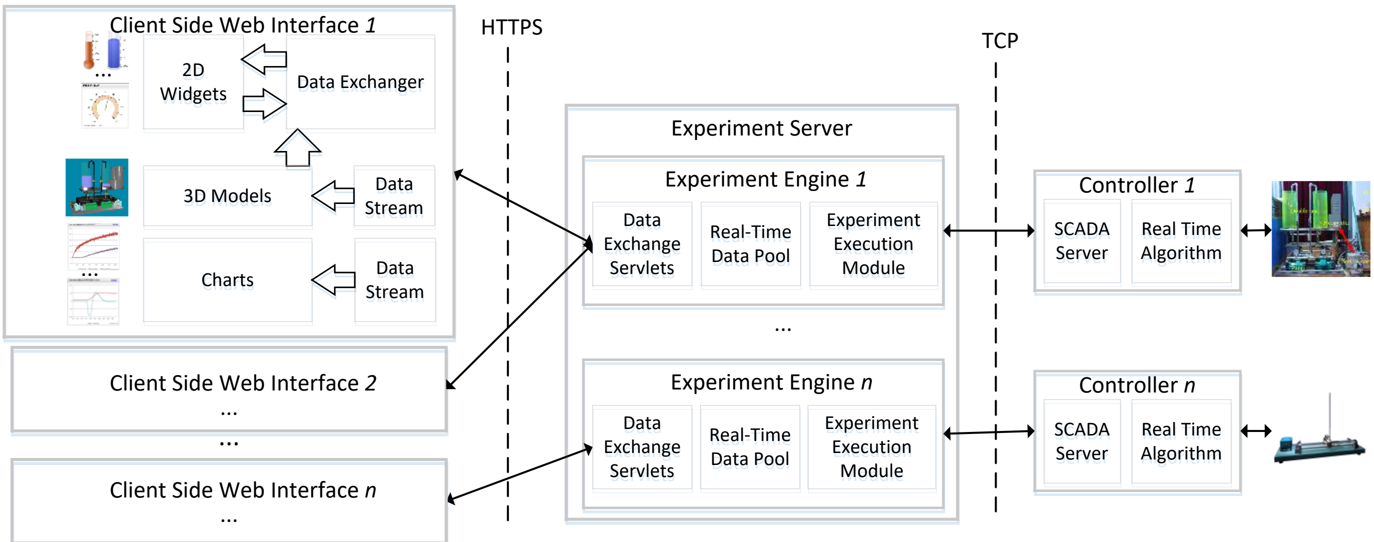


Fig. 8. Communication channels for NCSLab multi-tier architecture

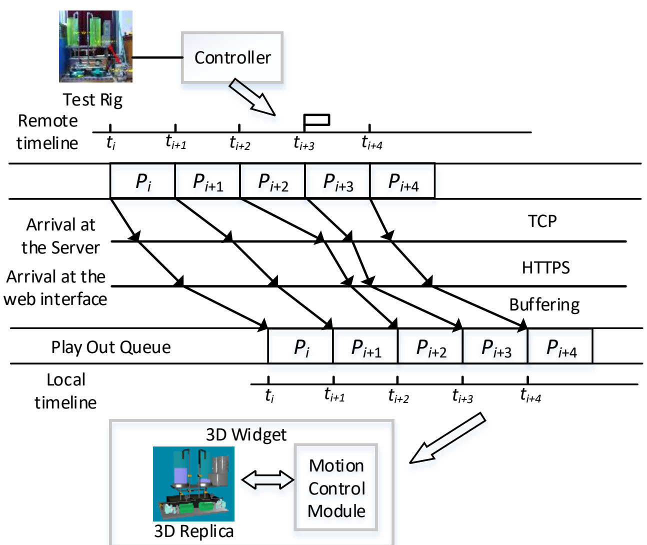


Fig. 9. Buffering algorithm for 3-D animation

experiment, the Motion Control Module retrieves the current local time using JavaScript method *getTime()* every cycle of the render loop. It requires the data of the "current time" from the *DataStream* object. The position of the ball and the angle of the lever arm are modified dynamically inside each render loop correspondingly as shown in Fig. 10.

As the data are retrieved from the buffer which is properly managed by the *DataStream* object, it is always available without the disturbance of the network uncertainty. From the user's point of view, it looks like that the 3-D animation of the ball and beam always changes following the practical test rig without any interruption and distortion.

Apart from "watching" the remote processes, the users are also allowed to interact with the 3D replicas. For example, by moving the cursor under the beam in Fig. 10, the set point of the ball position can be changed any time during the experiment.

VI. Teaching Practice Using NCSLab in Wuhan University

Online experimentation by NCSLab has been setup since 2011 in Wuhan University. In 2015, with dedicated laboratory space assigned, more experimental test rigs had been brought in. Currently, there are six practical test rigs and 20 virtual ones setup in the NCSLab of Wuhan University.

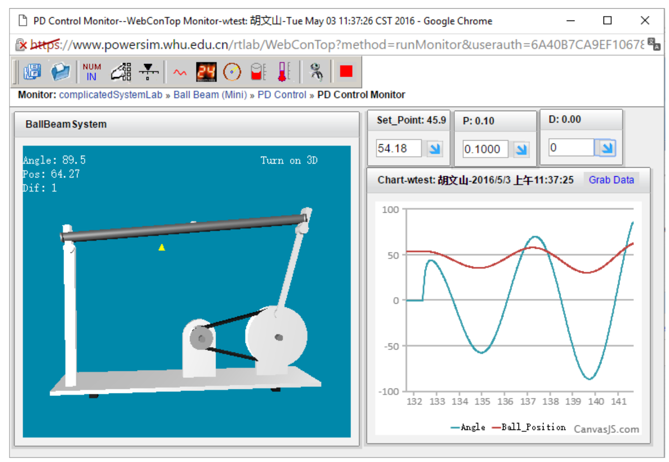


Fig. 10. Demonstration using ball and beam system

The application of online experimentation has been expanded to a large number of students since 2015. It has been used in three modules: *Classic Control Theory (CCT)*, *Control System Simulation and Computer Aided Design (CSS&CAD)* and *System Identification (SI)* by around 200 students per year. For each teaching module, the students are required to conduct several related experiments such as PID control, LQG control and system identification on the remote and virtual test rigs. Dedicated discussion groups using instant messaging software were setup to offer any help in case the students encounter difficulties.

1) Survey data analysis

As a regulation, all teaching modules are assessed online by the students. Their marks are important criteria for the university to judge the effectiveness of the modules. The general average marks for the three related modules is shown in Fig. 11. It can be seen that there was a strong boost on the student marks in 2015, as NCSLab was widely brought in that year.

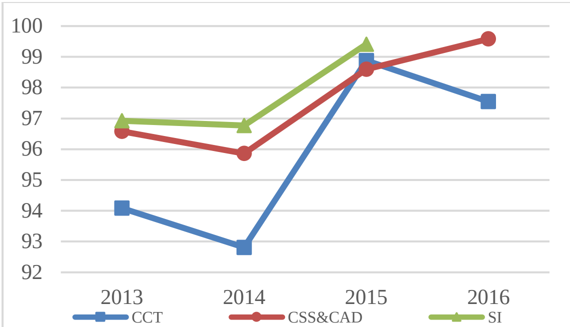


Fig. 11. Students' assessment (out of 100) for each module from 2013 to 2016. There was a boost in 2015 as NCSLab was brought in.

TABLE II
RESULTS OF SURVEY (SCALE 1-5)

	Question	2015(n=75) Mean	2015(n=75) S.D.	2016(n=78) Mean	2016(n=78) S.D.
Usability	Q1. NCSLab was easy to use?	4.84	0.717	4.56	1.191
	Q2. I was satisfied with the NCSLab system?	4.55	0.963	4.69	0.726
	Q3. I would recommend NCSLab to other people?	4.47	0.890	4.44	1.064
Educational Value	Q4. NCSLab was able to inspire my learning interest?	4.65	0.830	4.62	0.856
	Q5. NCSLab helped me to understand the course contents?	4.84	0.638	4.56	0.831
	Q6. NCSLab helped me to improve the laboratory skill?	4.76	0.803	4.38	1.131
Stability	Q7. NCSLab was working stable with no software bugs?	4.20	1.395	4.64	0.897
	Q8. There was no sense of time delay?	4.23	1.467	4.59	1.086

Apart from the university survey, students were also invited to complete questionnaires by the NCSLab team. There are mainly eight questions related to usability, educational values and stability. The results are shown in Table II.

Generally, the acceptance of NCSLab is notably high. Vast majority of the students thought it was able to help them understand the course content and also improve their laboratory skills. The stability could be improved. Even though the overall system is capable of working for months without any interrupt, some students did feedback the temporary malfunctioning of individual test rigs.

2) Traceable data analysis

NCSLab is able to trace and store the students' information during the experiments. The information includes when the students finish the online experiments, how much time they spend on the remote test rigs, and their images captured by the web cameras. These information are valuable for the assessment of the students' assignments.

Interesting phenomenon has been found according to the data. For each module, students were given plenty of time (normally one month). However, most of them chose to complete the experiments very close to deadline as shown in Fig. 12. The average time spending is listed in Table III. Regardless the preparation, only the time the students actually

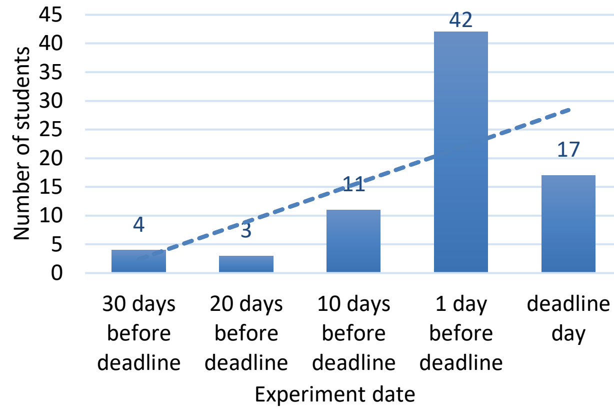


Fig. 12. Distribution of experiment date for module *Control System Simulation and Computer Aided Design* in 2016

TABLE III
AVERAGE TIME SPENDING OF STUDENTS

Module	Number of Experiments	Time Spending 2015	Time Spending 2016
Classic Control Theory	2(4 in 2016)	69 mins	191 mins
Control System Simulation and Computer Aided Design	5	215 mins	230 mins
System Identification	1	49 mins	56 mins

spent on the specific test rigs manipulating the devices is counted. The average time spending had increased in 2016, which can be explained by two reasons. Firstly, there were more complex experiments which requires the students to apply their own control algorithms in 2016. The second reason might be the deployment of HTML5 camera API which is able to record the experimenter' images. It was no longer possible for some students to invite their skilled classmates to do the experiments instead of them.

VII. TECHNICAL SUMMARY

The objective of the proposed NCSLab architecture is to build a powerful, secured, reliable and web-based 3-D online laboratory without any plug-ins. A technical analysis has been conducted and the characteristics of the system are summarized.

1) Cross-platform and Universality: NCSLab platform relies only on HTML5 technologies and no plug-in is required. Modern web browsers on both PC and mobile platforms, such as Google Chrome, Microsoft Edge, Internet Explorer, Mozilla Firefox and Safari are supported.

2) Graphical Interactivity: The experimental processes are presented to the users using virtual widgets such as real-time charts, gauges, and sliders.

3) 3-D interactivity: WebGL based 3-D visualization is also supported. The user is able to interact with the remote experimental processes through 3-D replicas.

4) User customized monitoring interface: NCSLab framework allows the user to customize their own monitoring interface using the 2-D and 3-D widgets.

5) User designed control algorithm: The system allows the users to upload their control algorithms and apply them to the remote controllers. The users are also allowed to customize the monitoring interface for their uploaded control algorithms.

6) Real-time communication schemes under web protocols: A buffered communication scheme based on HTTPS protocol is implemented to maintain the real-time synchronization among the remote controllers, servers and client sides.

7) Secured communication: Only HTTP and HTTPS protocols are required for NCSLab. HTTPS is implemented with higher security level for the communication during the online experiments.

8) Non-intrusive access: There is no installation required and the users do not have to give special permissions such as local file system access and privileged execution.

9) Client side image capture: The images of the experimenters are captured using the HTML5 Camera API, which can be used to verify the students' identities.

10) Services for every aspect of control experimentation: Online services are provided for the whole process of control experiments including remote monitoring, tuning, configuration and algorithm implementation.

By the creation of modular structure, multiple HTML5 toolkits with different functions have been successfully integrated seamless into the software architecture, which makes the cross-platform NCSLab capable of delivering equally powerful functionalities matching the level of plat-dependent applications.

VIII. Conclusion

This paper has introduced the architecture of NCSLab which is a universal, powerful, secured, reliable and plug-in free web-based 3-D online laboratory for control engineering education. The proposed system is able to deliver complex experimental services for university students covering the whole process of control engineering experimentation. Great flexibilities are provided to users as they are allowed to design and implement their own control algorithms, and also customize the remote monitoring interface. Various graphically interactive features such as virtual gauges, visual configuration of monitoring interface and real-time chart display are all achieved relying on HTML5 only. To enhance the sense of presence, NCSLab also reconstructs the experimental scenes in the form of 3-D visualization which are synchronized with remote control processes through the internet datalink. In order to get secured, reliable data exchange between the web interfaces and remote experimental processes, a multi-level real-time communication scheme based on HTTPS is also implemented. The new NCSLab system has been used in the teaching practice in Wuhan University. The feedbacks from both the survey and database data have verified the notably high effectiveness and general acceptance of NCSLab. This paper reveals the full detailed design and implementation of NCSLab architecture. Actually, the methodologies and technologies used for NCSLab could also be insightful for online laboratories in other areas to migrate towards universal and service oriented systems.

References

[1] A. Maiti, A. Kist, and A. Maxwell, "Real-time remote access laboratory with distributed and modular design," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3607–3618, Jun. 2015.

[2] A. Melkonyan, A. Gampe, M. Pontual, G. Huang, and D. Akopian, "Facilitating remote laboratory deployments using a relay gateway server architecture," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 477-485, Jan. 2014.

[3] L. Gomes and S. Bogosyan, "Current trends in remote laboratories," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4744–4756, Dec. 2009.

[4] T. de Jong, S. Sotiriou and D. Gillet, "Innovations in STEM education: the Go-Lab federation of online labs," *Smart Learn. Environ.*, vol. 1, no. 1, pp. 1-16, Dec. 2014.

[5] M. J. Callaghan, K. McCusker, J. L. Losada, J. Harkin, and S. Wilson, "Using game-based learning in virtual worlds to teach electronic and electrical engineering," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 575-584, Feb. 2013.

[6] V. Potkonjak, M. Gardner, V. Callaghan, P. Mattila, C. Guetl, V. M. Petrović, and K. Jovanović, "Virtual laboratories for education in science, technology, and engineering: A review," *Comput. Educ.*, vol. 95, pp. 309-327, Apr. 2016.

[7] L. Rodriguez-Gil, P. Orduna, J. Garcia-Zubia, I. Angulo, and D. Lopez-de-Ipina, "Graphic technologies for virtual, remote and hybrid laboratories: WebLab-FPGA hybrid lab," in *Proc. 11th Int. Conf. Remote Eng. Virtual Instrum., REV2014*, 2014, pp. 163-166.

[8] K. Henke, S. Ostendorff, H. Wuttke, and S. Simon, "Fields of applications for hybrid online labs," in *Proc. 10th Int. Conf. Remote Eng. Virtual Instrum., REV2013*, 2013, pp. 1-8.

[9] P. Orduna, P. H. Bailey, K. DeLong, D. Lopez-de-Ipina and J. Garcia-Zubia, "Towards federated interoperable bridges for sharing educational remote laboratories", *Comput. Human Behavior*, vol. 30, pp. 389-395, Jan. 2014.

[10] D. Lowe, H. Yeung, M. Tawfik, E. Sancristobal, M. Castro, P. Orduña, and T. Richter, " Interoperating remote laboratory management systems (RLMSs) for more efficient sharing of laboratory resources," *Comput Stand Interfaces*, vol. 43, pp. 21-29, 2016.

[11] J. Saenz, J. Chacon, L. De La Torre, A. Visioli, and S. Dormido, "Open and low-cost virtual and remote labs on control engineering," *IEEE Access*, vol. 3, pp. 805-814, Jun. 2015.

[12] C. S. Peek, O. D. Crisalle, S. Depraz and D. Gillet, "The virtual control laboratory paradigm: architectural design requirements and realization through a DC-motor example", *Int. J. Eng. Educ.*, vol. 21, no. 6, pp. 1134-1147, 2005.

[13] J. Sanchez, S. Dormido, R. Pastor, and F. Morilla, "A Java/Matlab-based environment for remote control system laboratories: Illustrated with an inverted pendulum," *IEEE Trans. Educ.*, vol. 47, no. 3, pp. 321-329, Aug. 2004.

[14] A. Barchowsky, B. M. Grainger, P. T. Lewis, A. D. Cardoza, G. F. Reed, and D. J. Carnovale, "Design and realization of an innovative workbench for electric power systems laboratories," *IEEE Trans. Power Syst.*, vol. 30, no. 6, pp. 2894-2901, Nov. 2015.

[15] P. N. A. Barata, M. R. Filho and M. V. A. Nunes, "Consolidating learning in power systems: Virtual reality applied to the study of the operation of electric power transformers," *IEEE Trans. Educ.*, vol. 58, no. 4, pp. 255-261, Nov. 2015.

[16] D. Ponta and G. Donzellini, "A virtual laboratory for digital design," *Int. J. Online Eng.*, vol. 8, pp. 5-11, 2008.

[17] Z. Nedic, "Demonstration of collaborative features of remote laboratory Netlab," *Int. J. Online Eng.*, vol. 9, pp.10-12, 2013.

[18] P. T. Goeser, W. M. Johnson, F. G. Hamza-Lup and D. Schaefer, "View – a virtual interactive web-based learning environment for engineering," *Adv. Eng. Educ.*, vol. 2, no. 3, pp.1-24, 2011.

[19] J. J. Rodriguez-Andina, L. Gomes, and S. Bogosyan, "Current trends in industrial electronics education," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3245–3252, Oct. 2010.

[20] T. Richter, R. Watson, S. Kassavetis, M. Kraft, P. Grube, D. Boehringer, P. de Vries, E. Hatzikraniotis, and S. Logothetidis, "The WebLabs of the University of Cambridge: A study of securing remote instrumentation," in *Proc. 9th Int. Conf. Remote Eng. Virtual Instrum., REV2012*, 2012, pp. 1-5.

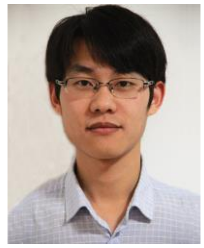
[21] A. Cardoso, D. Osório, J. Leitão, V. Sousa, and C. Teixeira, "A remote lab to simulate the physiological process of ingestion and excretion of a drug," *Int. J. Online Eng*., vol.12, no.4, pp. 74-76, Apr. 2016.

[22] H. Valdivieso, A. Mauricio, M. B. S. Sánchez,, D. A. U. Higuita,, R. C. Castelló, and M. Á. M. Villanueva, " Virtual laboratory for simulation and learning of cardiovascular system function in biomedical engineering studies," *Rev.Fac.Ing.Univ. Antioquia*, no.60, pp. 194-201, 2011.

[23] C. Martin-Villalba, A. Urquia, Y. Senichenkov, Y. Kolesov, "Two approaches to facilitate virtual lab implementation," *Comput. Sci. Eng.*, vol.16, no.1, pp.78-86, 2014.
[24] B. Scheucher, P. H. Bailey, C. Gutl, V. J. Harward, "Collaborative virtual 3d environment for internet-accessible physics experiments", *Int. J. Online Eng.*, vol. 5, pp. 65-71, Aug. 2009.
[25] D. Chaos, J. Chacón, J. Lopez-Orozco, and S. Dormido, "Virtual and remote robotic laboratory using EJS, MATLAB and LabVIEW," *Sensors*, vol. 13, no. 2, pp. 2595-2612, Feb. 2013.
[26] E. Besada-Portas, J. A. Lopez-Orozco, L. De La Torre, and J. M. de la Cruz, "Remote control laboratory using EJS applets and twinCAT programmable logic controllers," *IEEE Trans. Educ.*, vol. 56, no. 2, pp. 156-164, May. 2013.
[27] M. Tawfik, E. Sancristobal, S. Martin, R. Gil, G. Diaz, A. Colmenar, J. Peire, M. Castro, K. Nilsson, J. Zackrisson, L. Kansson, and I. Gustavsson, "Virtual instrument systems in reality (VISIR) for remote wiring and measurement of electronic circuits on breadboard," *IEEE Trans. Learn. Technol.*, vol. 6, no. 1, pp. 60-72, Jan-March. 2013.
[28] K. Bauer and L. A. Mendes, "Weblab of a control experiment in a newborn baby incubator," in *Proc. 12th Int. Conf. Remote Eng. Virtual Instrum., REV2015*, 2015, pp. 163-171.
[29] A. Yazidi, H. Henao, G.-A. Capolino, F. Betin, and F. Filippetti, "A web-based remote laboratory for monitoring and diagnosis of AC electrical machines," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4950–4959, Oct. 2011.
[30] J. Garcia-Zubia, P. Orduna, D. Lopez-de-Ipina, and G. R. Alves, "Addressing software impact in the design of remote laboratories," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4757-4767, Dec. 2009.
[31] L. Rodriguez-Gil, J. Garcia-Zubia, P. Orduna, and D. Lopez-de-Ipina, "Towards new multiplatform hybrid online laboratory models," *IEEE Trans. Learn. Technol.*, DOI 10.1109/TLT.2016.2591953.
[32] M. Tawfik, D. Lowe, C. Salzmann, D. Gillet, E. Sancristoba and M. Castro, "Defining the critical factors in the architectural design of remote laboratories," *IEEE Revista Iberoamer. Tecnol. Aprendizaje*, vol. 10, no. 4, pp. 269–279, Nov. 2015.
[33] D. A. H. Samuelsen, J. Bjørk, and O. H. Graven, "Converting a remote laboratory back end from remote panels in LabVIEW to HTML5," in *Proc. 12th Int. Conf. Remote Eng. Virtual Instrum., REV2015*, 2015, pp. 220-222.
[34] I. Titov, A. Glotov, I. Vlasov, and J. Mikolnikov, "Labicom labs 2015: remote laser virtual and remote lab, global navigation satellite systems virtual and remote lab, microwave amplifier remote lab," *Int. J. Online Eng.*, vol. 12, no.4, pp. 17-19, Apr. 2016.
[35] K. Henke, T. Vietzke, H.-D. Wuttke, and St. Ostendorff, "GOLDi – grid of online lab devices Ilmenau," *Int. J. Online Eng.*, vol. 12, no.4, pp. 11-13, Apr. 2016.
[36] M. Kaluz, J. Garcia-Zubia, M. Fikar and L. Cirka, "A flexible and configurable architecture for automatic control remote laboratories," *IEEE Trans. Learn. Technol.*, vol. 8, no. 3, pp. 299–310, Jul-Sep. 2015.
[37] F. Lustig, J. Dvorak, P. Kuriscak, and P. Brom, "Open modular hardware and softwarekit for creations of remote experiments accessible from PC and Mobile Platform," *Int. J. Online Eng.*, vol. 12, no.7, pp. 30-36, Jul. 2016.
[38] W. Hu, G.-P. Liu, D. Rees, and Y. Qiao, "Design and implementation of web-based control laboratory for test rigs in geographically diverse locations," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2343–2354, Jun. 2008.
[39] Y. Qiao, G. -P. Liu, G. Zheng and W. Hu, "NCSLab: A web-based global-scale control laboratory with rich interactive features," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3253-3265, Oct. 2010.
[40] W. Hu, G. -P. Liu and H. Zhou, "Web-based 3-D control laboratory for remote real-time experimentation," *IEEE Trans. Ind. Electron.*, vol. 60, no. 10, pp. 4673-4682, Oct. 2013.

Wenshan Hu received the B.S and M.Sc. degrees in control theory and applications from Wuhan University, Wuhan, China, in 2002 and 2004, respectively, and the Ph.D. degree in control engineering from the University of Glamorgan, Pontypridd, U.K., in 2008.

In June 2010, he joined Wuhan University, as an Associate Professor. His research interests include network-based control laboratories and wireless power transfer.



Zhongcheng Lei received the B.S. degrees in automation from Wuhan University, Wuhan, China, in 2014. He is currently working toward the Ph.D. degree at the Department of Automation, Wuhan University, Wuhan, China.

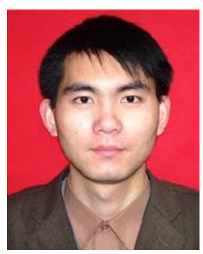
His current research interests include networked control systems and web-based remote laboratories.

Hong Zhou received the B.S. degree from the Central South University of Technology, Changsha, China, in 1982, the M.Sc. degree from Chongqing University, Chongqing, China, in 1988, and the Ph.D. degree in mechanical engineering from Wuhan University, Wuhan, China, in 2006.

He has been a Professor in Department of Automation, Wuhan University since 2000. His research interests include wireless power transfer, smart grid, and networked control systems.

Guo-Ping Liu (M'97-SM'99-F'11) received the B.Eng. and M.Eng. degrees in automation from the Central South University of Technology, Changsha, China, in 1982 and 1985, respectively, and the Ph.D. degree in control engineering from the University of Manchester Institute of Science and Technology (UMIST), Manchester, U.K., in 1992.

He is the Chair of Control Engineering with the University of South Wales, Pontypridd, U.K. He has been a Professor with the University of South Wales (formerly University of Glamorgan) since 2004. His research interests include networked control systems, nonlinear system identification and control, advanced control of industrial systems, and multiobjective optimization and control.



Qijun Deng received the B.S. and M.Sc. degrees in mechanical engineering and the Ph.D. degree in computer application technology from Wuhan University, Wuhan, China, in 1999, 2002, and 2005, respectively. In June 2005, he joined the Department of Automation, Wuhan University, where he is currently an Associate Professor. Hisresearch interests include wireless power transfer, distribution automation, and electrical power informatization.

Dongguo Zhou received the B.S. and Ph.D degree in College of Opto-electronic Engineering of Chongqing University, China, in 2008 and 2013 respectively. He is now a lecturer in Department of Automation, Wuhan University since 2014. His research interests include control application and signal processing.

Zhi-Wei Liu (M'14) received the B.S. degree from the Southwest Jiaotong University, Chengdu, China, in 2004, and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2011. He is currently an Associate Professor with the Department of Automation, School of Power and Mechanical Engineering, Wuhan University, Wuhan, China. His research interests include cooperative control of networked control systems.