Web-based 3D Interactive Virtual Control Laboratory Based on NCSLab Framework

http://dx.doi.org/10.3991/ijoe.v10i6.3845

W. Hu, H. Zhou, Z. -W. Liu and L. Zhong
Wuhan University, Wuhan, China

Abstract—**In this paper, the design and implementation of a web-based 3D virtual control laboratory with user interaction is introduced. The proposed system is based on the NCSLab framework, which has a six-tier distributed structure. Users are able to access diversely located resources through a unified user interface. In order to maximize users' sense of immersion during the virtual experiments, 3D models of practical test rigs are reconstructed in the users' web-browsers. The 3D images are synchronized with the remote real-time simulation on the server side through internet data links so the users are able to sense how the experiment processes are going on through the 3D animation. Moreover, the users are also allowed to interact with the virtual devices. They can operate on the virtual test rigs using the mice and keyboards, such as turning a switch on/off and changing the valve positions. These interactions are uploaded to the server side where the corresponding internal parameters of the real-time simulation are updated. The system implementation and deployment such as the 3D web-based interface, 3D modeling, mathematical modeling and internet based communication among the clients, servers and control units are explained in detail. At the end of the paper, a classic coupled tanks test rig has been selected as an example to illustrate how a given experimental process is integrated into the system.**

Index Terms—**Web and internet services, virtual reality, student experiments, control engineering education.**

I. INTRODUCTION

Experimentation is an important part of the teaching and learning process [1], especially for control engineering education. Apart from the concepts taught in lectures, experimentation enables students to experience the how the theories are applied to design practical systems, which could significantly enhance their understanding of the knowledge.

However, some experimental equipment could be expensive. Not all the educational institutions can afford all the resources they need, especially in developing countries. Without adequate experimentation, the quality of education could be severely affected.

With the rapid development of network technologies, remote experimentation [2] becomes an effective solution for the problem. Limited experimental resources are put online so they can be shared among institutions. Students are able to access these resources anywhere and anytime as long as they have internet access.

There have been various kinds of remote experimental instruments developed in recent years. In the area of control and automation, there have been many systems implemented successfully, such as DC servo motors [3], PLCs[4], inverted pendulums [5], robots [6][7], AC electric machines [8], 2DOF helicopters [9] and power electronics systems [10][11]. Various technologies and frameworks have been adopted in the development of the remote laboratories, such as Matlab, Java [5][12], Web2.0, Flash, iLab [13] and LabView [14], *etc*.

The early versions of the remote laboratories only concentrated on particular types of experimental devices with conventional server-client structure. With the development of the internet and cloud computing, some remote laboratories with more complex architectures have emerged [15]. These systems enable different kinds of test rigs diversely located in different institutions to be integrated together. WebLab-Deusto [16] and NCSLab [17-19] are examples of this kind of systems. Users are able to access these resources without considering their geographical locations. From the users' point of view, diversely located resources are put together under a unified web interface.

However, not all the resources are suitable for online access. It is highly possible that some expensive or fragile devices could be damaged by the online experiments. Unexpected interaction may result in hazardous consequences for the potentially dangerous experimental processes.

To avoid the risks, virtual experimentation [20-24] is an effective solution. There have been many systems developed for various courses such as digital design [25], microcontroller [26], measurement [22] and control engineering [20], etc. In [27], two software tools (Interactive and Rand Model Designer) were adopted to facilitate the implementation of a virtual laboratory. In [22], a JINI based distributed virtual laboratory framework called SimBAD is introduced. It offers services to teachers and students for live real laboratory experiences according "trial and error" and "problem solving approaches."

In virtual control laboratories, the experimental processes are modeled mathematically. The mathematical models rather than the practical test rigs are actually "controlled" and "monitored" during the experiments. As the physical equipment is no longer necessary, expensive or hazardous experimental processes can be placed online safely.

However, there is a major drawback for virtual laboratories. In most remote experiments, web cameras are used to provide real-time videos over the internet, which enables the users to actually observe how the experiments are going. However, this is not the case for virtual experimentation because the physical experimental processes do not exist. Without the help of real-time videos, the users can hardly experience the same sense of immersion as they do in hands-on experiments.

In recent years, web-based 3D technologies [28] have been widely used in many applications such as online games [29], urban management [30] and GIS. The 3D models of real experimental equipment can be imported and rendered in web browsers as a substitution for real-time videos. The users can “watch” experimental processes being reconstructed in the user interface. The 3D images can also display more details beyond the limitation of fixed web cameras. The virtual test rigs can be moved up/down and monitored from different angles without restrictions.

In this paper, a new web-based virtual 3D control laboratory with user interactivity is introduced. The new system is based on the distributed NCSLab framework. In order to provide the maximum sense of presence for the users, 3D virtual images that are synchronized with the remote real-time simulations are created in the user interface. Apart from watching the experimental process, the users are also encouraged to interact with the 3D virtual devices. They can “operate” on the virtual test rigs using mice and keyboards similar to the process in hands-on laboratories.

II. 3D INTERACTIVITY WITH NCSLAB FRAMEWORK

A. Distributed Structure of NCSLab Framework

The proposed virtual 3D control laboratory was built based on the NCSLab framework. NCSLab was originally designed for global scale web-based experimentation. It has a complex six-tier distributed architecture that enables the experimental resources diversely located in the different part of the world to be integrated together. Fig. 1 is the laboratory structure under the website www.ncslab.net. It has successfully integrated eight control test rigs located in four institutions from both China and the UK [17-19].

The distributed structure provides reliable real-time data exchange channels among the diversely located servers, web browsers and control units. The servers collect the real-time data from the control units and distribute these data to end users’ web browsers. The web browsers display these data in a web-based monitoring interface with many widgets, like dynamical charts, sliders, numeric inputs etc. The features of the NCSLab are listed below.

1) Virtual laboratory configuration: All the distributed test rigs are classified into several sub-laboratories by their functionalities rather than their locations. For example, all the flow control test rigs are put into the Flow Control Sub-laboratory and all the servo control test rigs are catalogued in the Servo Control Sub-laboratory, regardless of their geographical location.

2) Visual configuration of monitoring interface: NCSLab also gives users the freedom to customize their own monitoring interface, in which they can decide which signals should be observed and which parameters should be tuned during the experiments. Various widgets are provided for them.

3) Web-based control algorithm design: The users are encouraged to design their own control algorithms. There are two methods provided in NCSLab. First, the users can use Matlab Simulink to customize the algorithms and then upload them to NCSLab using a web-based interface. Second, NCSLab also provides a pure web-based GUI called WebconLink that enables the users to design algorithms by drawing block diagrams in their web browsers.

4) Remote and virtual experimentation: The NCSLab framework supports both remote and virtual experiments. If the control units are connected to the physical test rigs through hardware interfaces such as A/D and D/A converters, remote experimental services are provided. This is the case for *Control Unit A* and *Control Unit B* in Fig. 1. On the contrary, if the practical devices are replaced by mathematical models like *Control Unit C* and *Control Unit D*, the control units actually “control” the virtual “test rigs” represented by their mathematical models. In that case, the control units are working in the virtual experiment mode.

B. 3D Challenges for Virtual Experimentation

When all the control units are running in the virtual experiment mode, the whole system can be considered as a virtual laboratory. Fig. 2 is the welcome page of the proposed virtual laboratory in which twenty virtual control test rigs are integrated together in Wuhan University.

In web-based practical experiments, web cameras are setup in front of the test rigs. The real-time images of the experimentation process can be transferred to the users through the internet. However, this is not the case for virtual experiments. Due to the absence of practical devices, remote real-time images are no longer available. Without watching the real experiment process, the users can only interact with the web-based interface through 2D widgets such as charts, sliders, etc. It could be difficult for them to form the basic concepts of the experiments, such as how the experiments are actually conducted and what the practical experiments look like, especially for the beginners.

Therefore, it is a great challenge for virtual laboratories to deliver an equally strong sense of immersion without the support of real-time images. That is a main reason why virtual laboratories have received less preference compared to remote ones, even though they are much cheaper and safer.

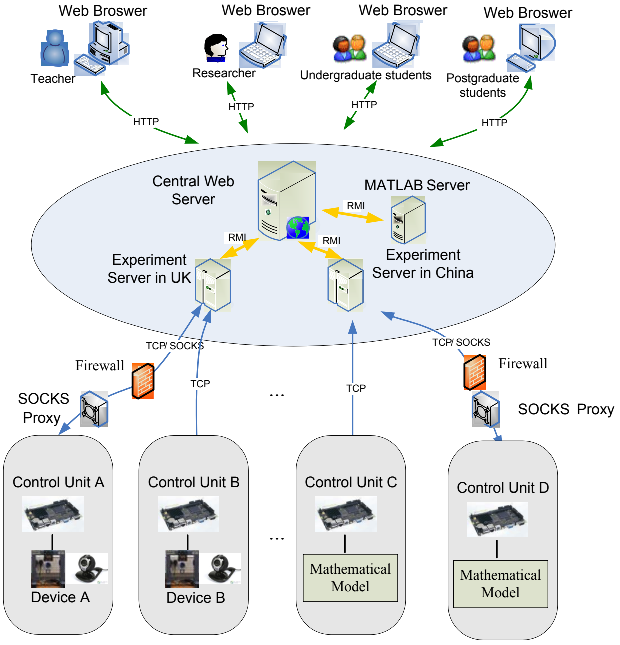


Figure 1. Structure of NCSLab

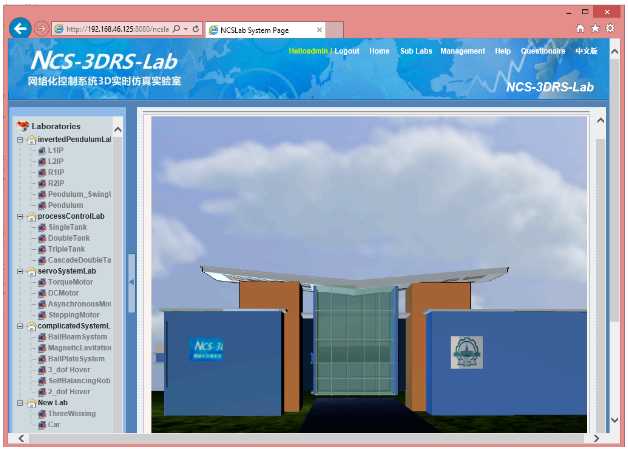


Figure 2. Welcome page of virtual laboratory

However, with the rapid development of 3D technology in recent years, 3D visualization could be an effective solution for the problem. The practical test rigs are modeled in 3D, and these 3D models are displayed in the users' web-based interface instead of in real-time videos. Through the data link between the user interface and remote control units, the 3D 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 similar to how they watch the live videos in remote experiments.

Moreover, the users are also able to interact with the 3D models using mice and keyboards with the support of 3D engines. They can rotate, zoom in/out and watch the test rigs from different angles. They can also operate on the 3D models like tuning a valve or turning on/off a switch similar to what they do in hands-on experiments. The parameter changes triggered by the interactivities are sent through the internet to the control units, where the internal parameters of the virtual experiment execution are updated immediately.

III. 3D SYSTEM DESIGN BASED ON NCSLAB

In order to achieve web-based interactive 3D virtual reality interface in the NCSLab framework, there are several necessary supporting technologies listed below.

A. 3D Modeling of the Experiment Devices

Experimental test rigs must be modeled in 3D before they can be imported into the web-based user interface. To give the maximum sense of presence, all the 3D models are built according to the real dimensions. There are many commercial software tools available for the 3D model design, such as 3DS Max, Solid Works and Pro/E *etc*.

In this paper, 3DS Max was selected as the 3D model development tool. The models designed in the 3DS Max were exported into the .3ds format, which can be parsed by Flash 3D engines. Fig. 3 is an example that shows the 3D model of a classic coupled tanks test rig being designed in a 3DS Max environment.

The complexity of these models must be considered carefully when designing the 3D models. The quality and the complexity have to be balanced. Even though the most powerful 3D engines with hardware acceleration are more than capable of processing millions of triangles, too complex models always result in big target files. In the internet environment, the long wait for downloading these files could be frustrating for the users.

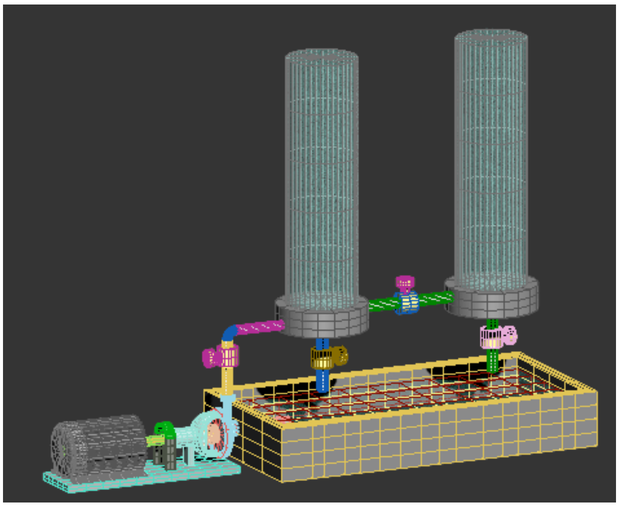


Figure 3. 3D model of a coupled tanks test rig in 3DS Max

B. Flash 3D Engine

In order to display the 3D models to the users, a powerful 3D engine is necessary. There are several web-based 3D solutions available in the market such as Adobe Flash, HTML5, and Google NaCl. Among them, Flash has the best penetration rate so far. Over 95% of the personal computers have installed Flash plug-ins. With the consideration of compatibility, Flash is selected as the 3D solution for the proposed virtual laboratory.

The early version of Flash only provides limited 3D capacities. Only relatively simple 3D applications can be rendered in web-based interface. The release of Flash 11 has changed the situation. Flash 11.0 supports hardware accelerated 3D rendering, which makes it possible to design more complex 3D applications. Stage 3D, which is a set of 3D API in Flash 11, enables advanced 3D capabilities on both PC and mobile platforms.

Based on the Stage 3D API, there have been several 3D engines released in recent years to help developers quickly design 3D web-based applications. These 3D engines include Proscenium 3D, Paperversion 3D and Away 3D, etc. Some of them are commercial software that requires license payment from the developers, and some of them are totally free.

Among them, Away3D is an open source 3D engine, written for the Adobe Flash platform in ActionScript 3, with full support of GPU acceleration. Using the resources provided by Stage 3D and Away 3D, 3D models can be easily imported into Flash Controls that are embedded into web browsers. Fig. 4 is the coupled tanks test rig rendered in 3D Flash Control.

C. 3D Interaction

With the support of the 3D engine, the developers are also capable of designing some codes supporting the interactivities between the users and 3D models. Away 3D provides rich resources that enable users to interact with the objects rendered in the 3D scenes. The users are allowed to use mice to pick up, drag, move, and rotate the 3D objects.

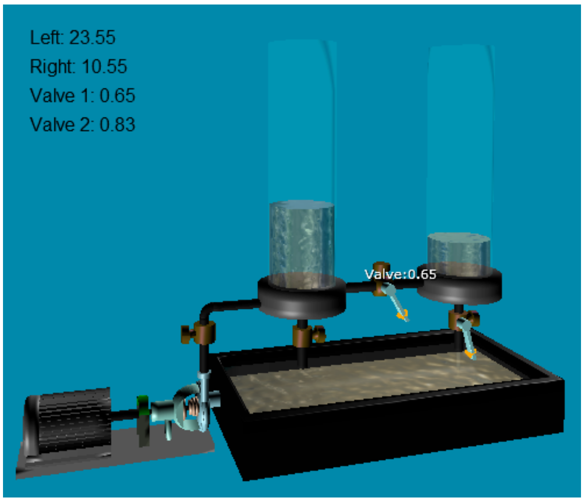


Figure 4. Virtual coupled tanks test rig rendered using Away 3D Engine with two interactive valves

Based on the APIs provided by Away 3D, the supporting codes for the user interactions are developed. By responding to the user's keyboard and mouse operations, the status of some components (like valves or switches) on the virtual test rigs can be modified.

For example, the virtual coupled tanks test rig in Fig. 4 has two valves. Apart from watching the real-time water level of the two tanks, the users are also allowed to use the mouse to pick up the handles and change valve positions. When the mouse operation is completed, the new valve position is converted into the corresponding parameter changes for the virtual experiment. An HTTP message is sent out to the control unit to update the internal parameters.

D. 3D Flash Embedded in Web Pages

The Flash Controls need to be imported into the web pages before they can be displayed in the users' web browsers. Fig. 5 shows how the Flash Controls and 3D models are embedded into the web-based interface.

On the server side, dynamic web pages (JSPs and Servlets) are generated by the Tomcat Web Server to setup the HTML structure according to user's experiment configuration stored in the MySQL database. These pages are downloaded to the users' web browsers and form the actual experimental user interface.

JavaScript codes are embedded in the web pages to guide the deployment of the 3D Flash Controls. These codes download the Flash .swf files from the web server and put them into the specified position according to the user's experimental configuration. When the Flash Controls are activated, corresponding 3D models are downloaded from the web server and rendered in the user interface.

The data exchanges between the 3D interface and the web server are bidirectional. On the one hand, the motions of the 3D models can be manipulated according to the real-time data collected from the control units through the web server, which realizes the synchronization between the animations and virtual experimental executions. On the other hand, the parameter changes triggered by the users' interactions can also be uploaded to the web server through HTTP connections.

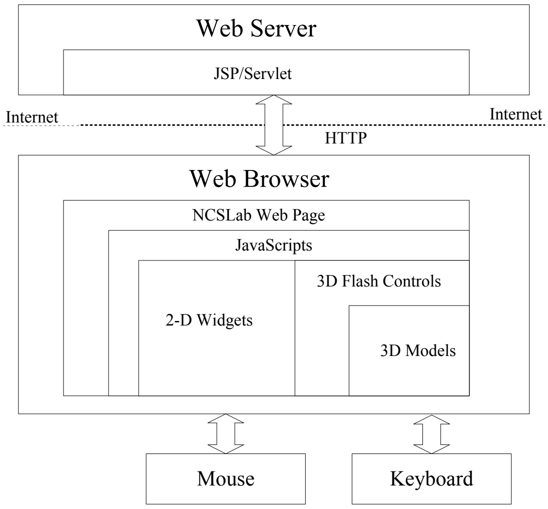


Figure 5. Structure of 3D web-based interface

IV. IMPLEMENTATION OF VIRTUAL LABORATORIES

A. Mathematical Models and 3D Models

All the control and real-time simulation programs executed in the control units are generated automatically using Matlab Real-Time Workshop (RTW). RTW is a powerful HIL (hardware in loop) tool that is able to convert the Simulink control block diagrams into C or C++ source codes. These source codes are compiled into executable programs using Arm Linux GCC complier. These programs can be downloaded into the control units and executed in the ARM based devices.

For a virtual experiment of a particular test rig, the mathematic model can be designed in the Simulink environment as a block diagram with multiple signals and parameters. The 3D models can also be built in the 3DS Max environment. In order to realize the synchronization between the mathematical model and the 3D model during the virtual experiments, a kind of interconnection between the two kinds of models is established.

For example, Fig. 6 shows the interconnections between the mathematical and 3D models for a coupled tanks platform. On the top of the figure is the Simulink block diagram of the coupled tanks control system. It can be seen that the mathematical model is represented in an S-Function module and is controlled by a PI controller.

In the Simulink diagram, there are two critical signals related to animation of the 3D model. "Water Level 1" and "Water Level 2" represent the water levels for the left and right tanks, respectively. There are also two parameters related to the user interactions. The valve positions in the 3D models are linked with the internal parameters of the two input blocks in the Simulink.

B. Data Links between the Mathematical Models and 3D Models

When the users are conducting virtual experiments, the 3D virtual test rigs displayed in the client side user interface and the virtual experiments executed in the remote control units are synchronized through network data links provided by the NCSLab framework. A bidirectional data exchange scheme is established as shown in Fig 7.

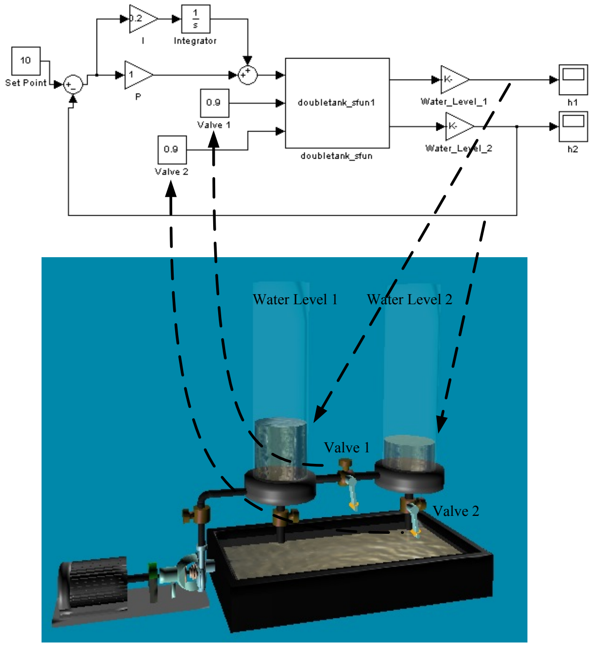


Figure 6. Interconnections between the mathematical model and 3D model for a coupled tanks test rig

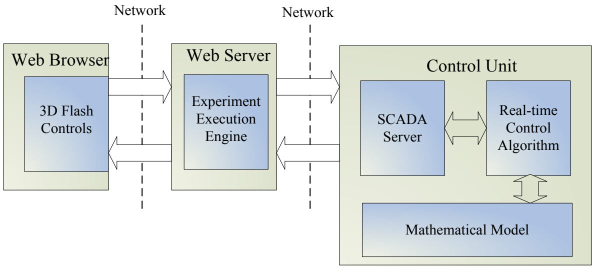


Figure 7. Structure of the data link between the 3D Flash Control and real-time simulation

All the programs executed in the control units for NCSLab provide networked data exchange services. All the internal signals and parameters can be accessed through a group of APIs provided by RTW.

The Experiment Execution Engine in the web server exchanges the data with the control units through the network following the Netcon Supervisory Protocol [31]. As soon as an algorithm is downloaded to the control unit, the experiment execution engine connects to the SCADA service and starts to retrieve real-time data.

When a user starts to monitor the experimental process, the connection between the web browser and the web server is established automatically. The real-time data of the interconnected signals are collected by the 3D Flash Controls, which perform the corresponding actions on the dynamical 3D models.

In the meantime, user interactions are converted into the corresponding parameter changes for the remote virtual experimental process. Network messages are generated by the 3D Flash Controls and sent to the corresponding control unit through the internet data links. When the control unit receives the messages, the corresponding parameters are updated immediately.

For example, during the coupled tanks control experiments, the water levels for both tanks in the user interface are synchronized with the real-time calculation of the corresponding signals “Water Level 1” and “Water Level 2” all the time through the internet data link. The users are also allowed use the mouse to change the valve positions on the 3D model. When the interactions occur, the new valve positions are sent back to the control unit, where the internal parameters associated with the valve positions are updated to the new values.

Under the internet environment, connection speed is always a big issue, which affects the performance of online laboratory greatly. The network delays among the control units, web servers and client side user interfaces may change frequently. The momentary variant of time delay is very harmful as it may alter the sequence of the data, which distorts the 3D animation.

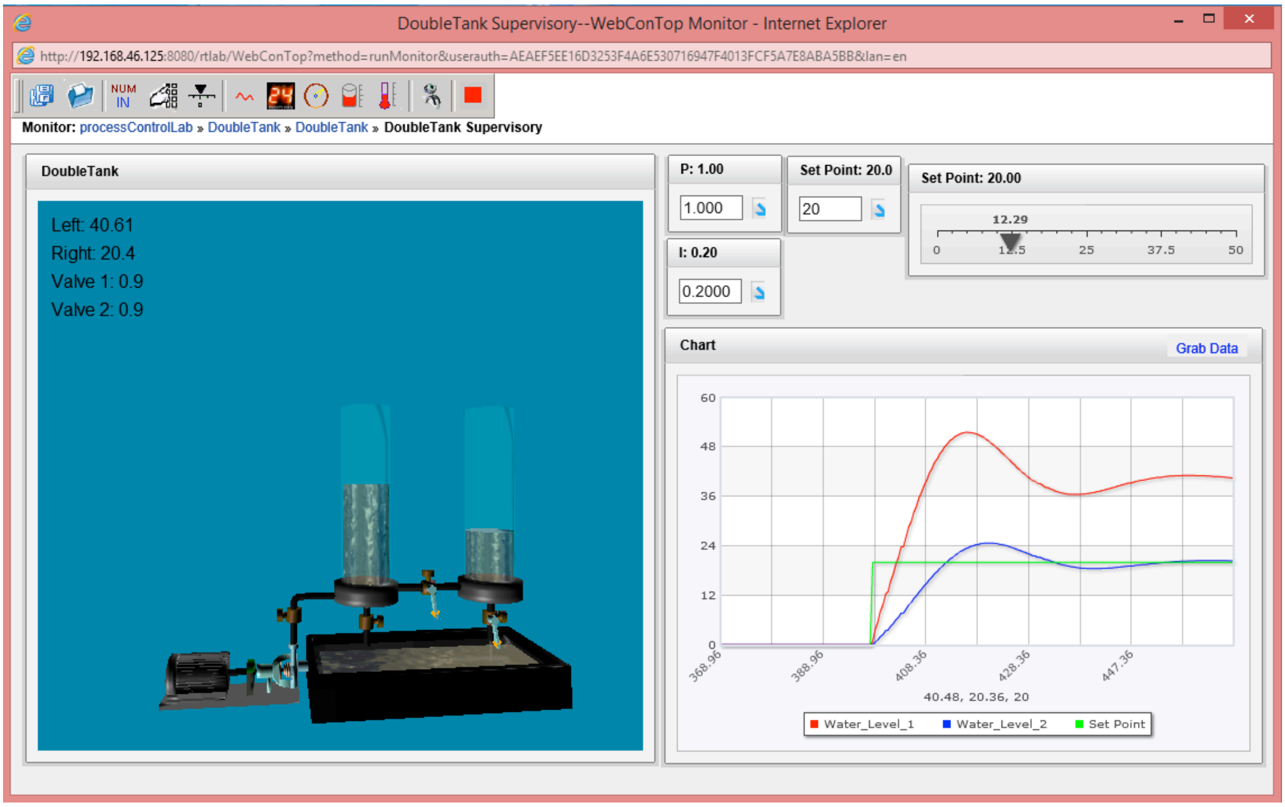
In order to get the 3D reconstruction free from the disturbance of random time delay, a simple buffering algorithm is designed. All the data are transmitted with a timestamp and are buffered on the client side before being “played” for the 3D animation. The algorithm always keeps a certain amount of latest data stored in the buffer. Therefore, small random changes of the network delay won’t affect the 3D animation.

C. *Web-based 3D Interface with Interactivities*

As a feature of NCSLab, users are allowed to customize their own monitoring interface. There are many widgets available for them such as sliders, angular gauges and thermometers. These widgets are connected to the internal signals and parameters of the remote virtual experiments. The users can drag them from the toolbox and place them anywhere they want in the user interface.

Apart from the 2D components, the 3D visualization proposed in this paper gives the users a vivid sense of what the experiments look like and how the experiments are going, even though the physical experiment processes do not exist and all the data are retrieved from the real-time simulation. By properly designing the mathematical and 3D models, most of the properties of the practical experimentation can be simulated and represented remotely to the users in the web-based virtual reality interface.

Fig. 8 is an example of a customized web-based 3D virtual experimental interface for the coupled tanks control test rig. The 3D images and 2D widgets are provided to the users together. The users are able to “watch” the real-time water levels from the 3D Flash Control on the right hand side. They can also tune the valves on the virtual test rig to change the “water flows.” Apart from the 3D interface, there are also a few 2D widgets provided for the users. They can adjust the PI control parameters and change the set point of the control system. Moreover, the real-time data of the two water levels and set point are plotted in a dynamic chart.



DoubleTank Supervisory--WebConTop Monitor - Internet Explorer

http://192.168.46.125:8080/rtlab/WebConTop?method=runMonitor&userauth=AEAEF5EE16D3253F4A6E530716947F4013FCF5A7E8ABA5BB&lan=en

Monitor: processControlLab » DoubleTank » DoubleTank » DoubleTank Supervisory

DoubleTank

Left: 40.61
Right: 20.4
Valve 1: 0.9
Valve 2: 0.9

P: 1.00 — 1.000

Set Point: 20.0 — 20

I: 0.20 — 0.2000

Set Point: 20.00

Figure 8. Web-based 3-D virtual experimental interface for coupled tanks test rig

D. Design Flow

The typical design flow to integrate a specific virtual experimental process is summarized in Fig. 9. The mathematical modeling and 3D design are two parallel procedures.

1) Mathematical modeling: The mathematical models are built as a Simulink block diagram. They are compiled into executable codes using Real-time Workshop.

2) 3D design: The 3D models are designed in a 3DS Max environment. Using Flash Builder with an Away 3D library, 3D Flash Controls are developed to create the 3D scenes inside web browsers and also process the user interactivities.

When the mathematical modeling and 3D design are completed, a new experimental process is registered in NCSLab and a new control unit is allocated for the virtual experiment. The executable simulation codes are stored into the database in the web server. It will be downloaded to the control unit when the user starts the virtual experiment. Supporting web pages will also be developed, as the 3D Flash Controls can be embedded into the web-based interface.

E. Current Implementation

Currently, twenty different control experimental systems including coupled tanks, inverted pendulums and servo motors, etc. are integrated into the proposed 3D virtual control laboratory. The main server is deployed from the campus of Wuhan University, China.

The virtual test rigs are cataloged into several sub-laboratories such as process control laboratory, servo control laboratory and complex system laboratories, etc. according to their functionalities. Fig. 10 shows snapshots of the virtual experimental processes for the ball and beam control system and the inverted pendulum system. They can be accessed through the web address http://control.whu.edu.cn.

If users want to perform experiments on a particular kind of test rig, they can search for it in the corresponding sub-laboratory. NCSLab has a distributed structure that supports the servers and control unit diversely located in different physical locations. However, from the users' point of view, they don't have to find out where the virtual experimental facilities are actually located. They only need to log onto the website and all the resources are accessible for them.

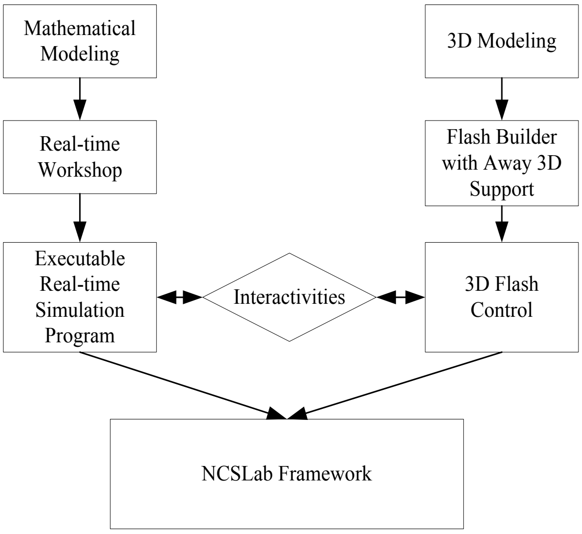


Figure 9. Design and implementation flow of a specific virtual experimental process

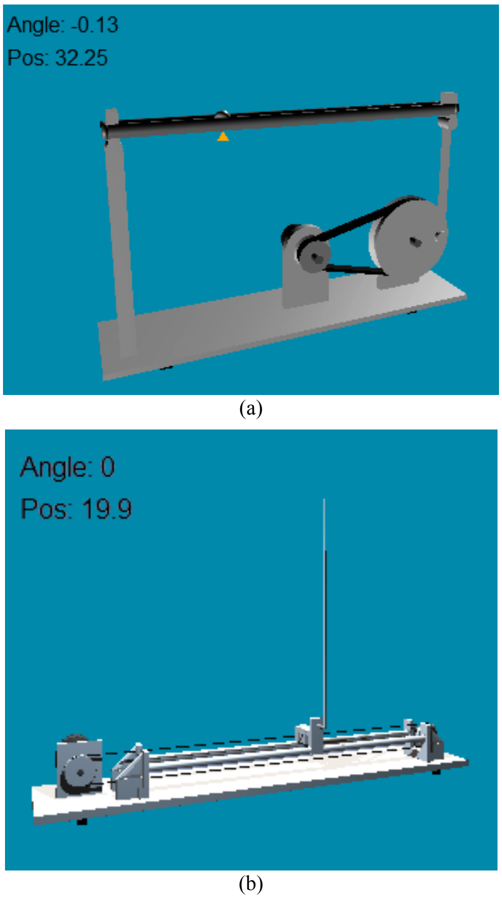


Figure 10. Snapshots of virtual experiments. (a) Ball and beam control system. (b) Inverted pendulum system

V. COUPLED TANKS SYSTEM - AN EXAMPLE

As the coupled tanks system has become a “design classic” for control engineering education, it was selected as the example to illustrate how experiments are integrated into the virtual laboratory. There is a coupled tanks system whose mathematical model is represented as

$$h_1 = (K_m u - a_1\sqrt{2g|h_1 - h_2|})/A \tag{1}$$

$$h_2 = (a_1\sqrt{2g|h_1 - h_2|} - a_2\sqrt{2gh_2})/A \tag{2}$$

$$a_1 = k_1 a_0 \tag{3}$$

$$a_2 = k_2 a_0 \tag{4}$$

where h_1 and h_2 are the water levels of the two tanks, u is the input power of the pump, K_m is the flow coefficient, g is the gravity constant, A is the cross section area of the tanks, a_0 is the cross section area of the pipes and k_1 and k_2 are the positions of the two valves. In order to describe the given mathematical model in Simulink, an S-function module was designed as shown in Fig. 11. The power of the water pump and the positions of the two valves are considered as the inputs. The water levels of the two tanks are considered as the outputs. C codes are programmed to represent the differential equations (1-4) in the S-function module.

To regulate the water level of the second tank, the system is controlled by a closed loop PI controller. The overall system, including both the mathematic model and controller, is represented in the Simulink block diagram shown in Fig. 6.

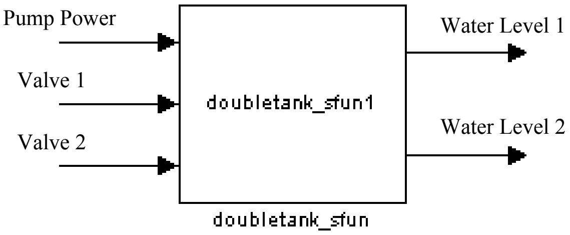


Figure 11. S-Function module for the coupled tanks mathematical model

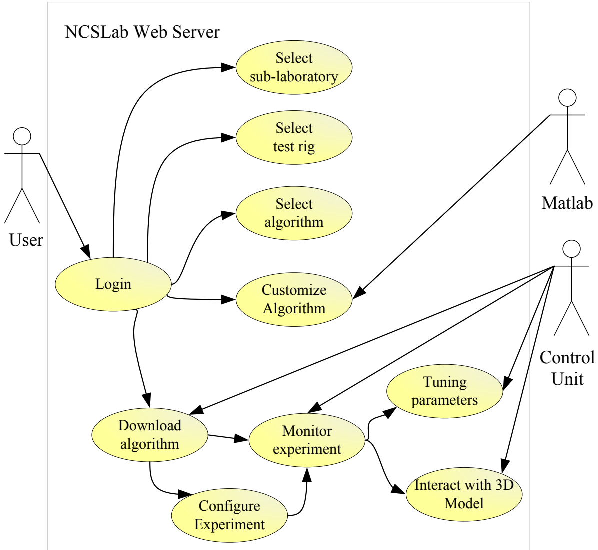


Figure 12. Use Case Diagram for virtual experimentation

Apart from the mathematical model, the 3D model of the practical test rig was built in 3DS Max, as shown in Fig. 3. Both the 3D model and the mathematic model are imported into the database of the virtual laboratory. A dedicated ARM-based control unit is deployed to implement actual remote real-time simulation.

Fig. 12 is the Use Case Diagram for virtual experimentation in NCSLab. To conduct experiments on the coupled tanks control system, users can log onto the system using their web browser. All the experimental facilities, which are cataloged into several sub-laboratories, are listed. The coupled tanks system belongs to the Process Control Sub-laboratory. By clicking the corresponding web link, the browser is directed to the web pages for the coupled tanks, where all the detailed documentation about the test rig, such as the manual and mathematical model, are provided to the users.

For a given virtual experimental test rig, there are several control algorithms provided for the users such as PID and LQR, etc. Apart from that, the users are also allowed to customize their own algorithms. The detailed implementation of customized control algorithms is introduced in [18]. Generally, the users have two following choices.

1) They can download an MDL template that contains the mathematical model of the coupled tanks and complete the algorithm design using Simulink. In this case, the users need to install Matlab on their PCs. They are encouraged to test the customized algorithms in the Matlab environment before uploading them to NCSLab.

2) They can also design their control algorithms directly using the web-based design tool Netconlink [18], which is provided by the NCSLab Framework with a user interface similar to Simulink.

Both the default and customized algorithms are listed for the users. They can pick any one they like and download it to the remote control unit, which initiates the remote virtual experiment.

Using the visual configuration tools provided by the NCSLab framework, the users are able to place both the 3D models and 2D widgets freely in the user interface. When the visual configuration process is complete, they can start to monitor the experimental process by clicking on the "run" button in the toolbox bar. Then the 3D experimental process is represented to the users with the interface, as shown in Fig. 8. They can monitor the 3D experimental process and tune the parameters. Moreover, they are also allowed to virtually interact with the "test rig" similar to what they do in hand-on experiments.

VI. CONCLUSION

In this paper, the design of a web-based virtual 3D interactive control laboratory has been introduced. The use of virtual experimentation can not only avoid the potential risks of the fragile or hazardous equipment, but also greatly reduce the costs. The proposed system is based on the NCSLab framework. Apart from the distributed architecture that allows the experimental resources diversely located in different places to be integrated together under a single website, the new virtual laboratory provides web-based 3D virtual experimental services for the users. The users are able to "watch" the experimental processes through the 3D animation of virtual test rigs reconstructed in their web browser. Moreover, they can also interact with the virtual equipment using mice and keyboards similar to how they interact with the practical test rigs in hands-on experiments. Even though the physical experimental processes do not exist, the users can still receive a strong sense of presence.

The detailed design and implementation of the proposed system have been represented in this paper, including the 3D modeling, mathematical modeling, 3D rendering and 3D interactivity. An example of a "classic" coupled tanks system has also been provided to illustrate how a practical experimental process is integrated into the virtual laboratory.

The future plans for the NCSLab project have been proposed as follows.

*1) Dealing with unexpected interactions.*Unpected interactions from the users could trigger unexpected results, which negitvely affect the user experience. If the users are doing something wrong or unexcpected, the system should be able to detect the wrongdoings and provide warnings.

*2) Active time delay compensation.*The random time delay among the control units, servers and web broswers could disturb the 3D reconstruction in the user interface. An active time delay compensation algorithm needs to be designed to minimize the latency between the virtual experiment executation and 3D replay.

3) User feedback analysis. The proposed interactive virtual laboratory was planned to be applied to the practical teaching activities at Wuhan University. Student feedback will be collected and analyzed. These first hand data could be very valuable to evaluate the effectiveness of the virtual experiments compared with the hands-on and remote ones.

REFERENCES

[1] L. Gomes, and S. Bogosyan, "Current trends in remote laboratories," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 12, pp. 4744-4756, Dec 2009 http://dx.doi.org/10.1109/TIE.2009.2033293

[2] S. Seiler, "Current Trends in Remote and Virtual Lab Engineering. Where are we in 2013? " *International Journal of Online Engineering*, vol. 9, pp. 12-16, 2013 http://dx.doi.org/10.3991/ijoe.v9i6.2898

[3] G. Sziebig, B. Takarics, and P. Korondi, "Control of an embedded system via Internet," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3324-3333, Oct 2010 http://dx.doi.org/10.1109/TIE.2010.2041132

[4] Y. S. Kim and H. M. Kim, "Design of a New Virtual Interaction Based PLC Training Using Virtual Sensors and Actuators: System and Its Application," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013

[5] 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 Transactions on Education*, vol. 47, no. 3, pp. 321–329, 2004. http://dx.doi.org/10.1109/TE.2004.825525

[6] R. Marin, P.J. Sanz, P. Nebot, R. Wirz, "A multimodal interface to control a robot arm via the web: a case study on remote programming," *IEEE Trans. on Industrial Electronics,* vol. 52, no. 6, pp. 1506- 1520, Dec 2005http://dx.doi.org/10.1109/TIE.2005.858733

[7] D. Lopez, R. Cedazo, F. M. Sanchez, J. M. Sebastian, "Ciclope Robot: Web-Based System to Remote Program an embedded Real-Time Sys," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 12, pp. 4791-4797, Dec 2009 http://dx.doi.org/10.1109/TIE.2008.2007007

[8] A. Yazidi, H. Henao, G.-A. Capolino, F. Betin, F. Filippetti, "A Web-Based Remote Laboratory for Monitoring and Diagnosis of AC Electrical Machines ," *IEEE Trans. on Industrial Electronics,* vol. 58, no. 10, pp. 4950-4959, Oct 2011. http://dx.doi.org/10.1109/TIE.2011.2109331

[9] C. C. Ko, B. M. Chen, J. P. Chen, J. Zhang, and K. C. Tan, "A Web-based laboratory on control of a two-degrees-of-freedom helicopter," *International Journal of Engineering Education*, vol. 21, no. 6, pp. 1017–1030, 2005.

[10] S. C. Wang and Y. H. Liu, "Software-reconfigurable e-learning platform for power electronics courses," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2416-2424, Jun 2008http://dx.doi.org/10.1109/TIE.2008.922592

[11] P. Bauer, J. Dudak, D. Maga, and V. Hajek, "Distance practical education for power electronics," *International Journal of Engineering Education*, vol. 23, no. 6, pp. 1210-1218, 2007.

[12] G. Farias, R. De Keyser, S. Dormido, F. Esquembre, "Developing Networked Control Labs: A Matlab and Easy Java Simulations Approach ," *IEEE Trans. on Industrial Electronics,* vol. 57, no. 10, pp. 3266-3275, Oct 2010 http://dx.doi.org/10.1109/TIE.2010.2041130

[13] V. J. Harward, J. A. Del Alamo, S. R. Lerman, P. H. Bailey, J. Carpenter, K. DeLong, C. Felknor, J. Hardison, B. Harrison, I. Jabbour, P. D. Long, T. Mao, L. Naamani, J. Northridge, M. Schulz, D. Talavera, C. Varadharajan, S. Wang, K. Yehia, R. Zbib, and D. Zych, "The iLab shared architecture: A Web services infrastructure to build communities of Internet accessible laboratories," *Proceedings of the IEEE*, vol. 96, no. 6, pp. 931–950, Jun. 2008. http://dx.doi.org/10.1109/JPROC.2008.921607

[14] A. Leva, and F. Donida, "Multifunctional Remote Laboratory for Education in Automatic Control: The CrAutoLab Experience," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2433-2438, Jun 2008 http://dx.doi.org/10.1109/TIE.2008.922590

[15] A.G. Vicente, Muñ, I.B. oz, J.L.L. Galilea, P.A.R. del Toro, "Remote Automation Laboratory Using a Cluster of Virtual Machines ," *IEEE Transactions on Industrial Electronics,* vol. 57,

no. 10, pp. 3276-3283, Oct 2010 http://dx.doi.org/10.1109/TIE.2010.2041131

[16] L. Gomes and J. García-Zubía, *Advances in Remote Laboratory and e-Learning Experiences*. Bilbao, Spain: Univ. Deusto, 2007.

[17] W. Hu, G. P. Liu, D. Rees, and Y. Qiao, "Design and implementation of Web-based control laboratory for experiment devices in geographically diverse locations," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2343–2354, 2008. http://dx.doi.org/10.1109/TIE.2008.920646

[18] Y. Qiao, G. P. Liu, G. Zheng and W. Hu, "NCSLab: A web-based global-scale control laboratory with rich interactive features," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3253-3265, Oct, 2010. http://dx.doi.org/10.1109/TIE.2009.2027924

[19] W. Hu, G. P. Liu and H. Zhou, "Web-Based 3D Control Laboratory for Remote Real-Time Experimentation," ," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 10, pp. 4673-4682, Oct, 2013 http://dx.doi.org/10.1109/TIE.2012.2208440

[20] S. Uran and K. Jezernik, "Virtual laboratory for creative control design experiments," *IEEE Transaction on Education*, vol. 51, no.1 pp.69-75, 2010 http://dx.doi.org/10.1109/TE.2007.906599

[21] D. S. Gubsky, V. V. Zemlyakov and I. V. Mamay, "The Modern Approach to Virtual Laboratory Workshop," *International Journal of Online Engineering*, vol. 10, pp. 56-59, 2014 http://dx.doi.org/10.3991/ijoe.v10i2.3496

[22] F. Colace, M. De Santo, L. Greco, "SimBAD: a Virtual Lab based on the Jini Framework, " *International Journal of Online Engineering*, vol. 10, pp. 22-29, 2014 http://dx.doi.org/10.3991/ijoe.v10i2.3210

[23] 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", *International journal of engineering education*, vol. 21, no. 6, pp. 1134-1147, 2005

[24] E. Granado, W. Colmenares, M. Strefezza and A. Alonso, "A Web-Based Virtual Laboratory for Teaching Automatic Control," *Computing Applications in Engineering Education*, vol. 12, pp 192-197, 2005

[25] D. Ponta and G. Donzellini, "A Virtual Laboratory for Digital Design," *International Journal of Online Engineering*, vol. 8, pp. 5-11, 2008

[26] S. Seiler, R. Sell, D. Ptasik and M. Bölter, "Holistic Web-based Virtual Micro Controller Framework for Research and Education," *International Journal of Online Engineering*, vol. 8, pp. 58-64, 2012

[27] C. Martin-Villalba, A. Urquia, Y. Senichenkov, Y. Kolesov, "Two Approaches to Facilitate Virtual Lab Implementation," *Computing in Science & Engineering* , vol.16, no.1, pp.78-86, 2014 http://dx.doi.org/10.1109/MCSE.2014.29

[28] J. García-Zubia, J. Irurzun, I. Angulo, P. Orduña, J. Ruiz-de-Garibay, U. Hernández, M. Castro and , E. Sancristobal, "Second-Lab: A Remote Laboratory under Second Life," *IEEE Education Conference*, 2010

[29] C. Carter, A. E. Rhalibi, M. Merabti, "Development and Deployment of Cross-Platform 3D Web-based Games," *Developments in E-systems Engineering (DESE)*, pp. 149 – 154, London, 2010

[30] F. Lamberti, A. Sanna, and E. A. H. Ramirez, "Web-based 3D visualization for intelligent street lighting," Proceedings of the 16th International Conference on 3D Web, pp. 151-154, New York, 2011

[31] Y. Zhu, G. Zheng, Z. Dong, and G. P. Liu, "Design of Supervisory Software for Ethernet-Based Control Systems," *IEEE International Conference on Networking, Sensing and Control*, pp. 36-41, London, 2007

AUTHORS

W. Hu is with Department of Automation, School of Power and Mechanical Engineering, Wuhan University, 430072, China. (e-mail: wenshan.hu@whu.edu.cn)

H. Zhou is with Department of Automation, School of Power and Mechanical Engineering, Wuhan University, 430072, China. (e-mail: hzhouwuhee@whu.edu.cn)

Z. –W. Liu is with Department of Automation, School of Power and Mechanical Engineering, Wuhan University, 430072, China. (e-mail: liuzw@whu.edu.cn)

L. Zhong is with Department of Automation, School of Power and Mechanical Engineering, Wuhan University, 430072, China. (e-mail: zhongliang@whu.edu.cn)

The work was supported by the National Science Foundation of China under Grant 61374064. Submitted 08 May 2014. Published as resubmitted by the authors 25 October 2014.