RemoteLab – an E-mail Based On-Line Control Experiment Service

Hung-Ming Cheng, George T.-C. Chiu, and Huei Peng

Abstract— **The Internet provides a new environment of developing a variety of applications for both educational and research purposes. At Purdue University, we have developed an innovative remote-access control-engineering laboratory. Remote control, network reliability, and safety features are integrated into the experimental hardware and software design. Remotely located students are able to develop and run controllers on the experiments in the laboratory, and they can effectively use the laboratory from anywhere over Internet. The service has been successfully tested in a graduate level course at University of Michigan as well as in a distance learning course with the Purdue Continuing Engineering Education Program.**

I. Introduction

It is well accepted that hands-on experience is essential in control education. Starting in the 1980s, education institutions have strived to develop and maintain undergraduate and graduate control related laboratories to provide hands-on experience. However, the maintenance of laboratories is not only expensive, but it also requires constant attention. When the courses are offered, teaching assistants need to be employed to conduct laboratory sessions. Laboratory instrumentations need to be constantly maintained, and software licenses also have to be updated. In many cases, the hardware and software evolutions do not coincide, which creates more issues. When the academic institutions are under financial pressure, laboratory maintenance is one of the easier options to reduce the cost. As academic institutions upgrade their control hands-on experience, it would be ideal if the institutions can also leverage the investment. Therefore, instead of sharing files, we would like to share laboratory/experiment resource.

The Internet has accelerated the availability of distant-learning environment [1][2]. Attempts of bringing hands-on experience to distance learning have also been investigated by many researchers [3]-[7]. Early in the 1960s, scientists had started to develop the paradigms of remotely operated experiments. In 1967, Ferrell and Sheridan [8] proposed the supervisory control scheme for the long distance transmission delay. Kondraske et al. [9] had also investigated a cost-effective way to conduct collaborative work by a distributed laboratory approach through the sharing of resources. In the 1990s, the concept of network accessible laboratory became popular [10][11]. Argonne National Laboratory provides an on-line microscope laboratory [12] over the Internet for both educational and research purposes. It gives an option for providing some form of hands-on experience for distance learning students.

The e-mail based on-line control experiment service, which will be referred as the *RemoteLab*, is developed to provide a cost-effective laboratory experience for on-campus students, and it can be easily ported to provide near hands-on experience for distance learning students. Most of the existing remote laboratories are based on a web server structure, where the users access a web page to interact with the laboratory features. As a web server, the site is vulnerable to potential intruders. Most sites require the use of tools, such as CGI and database, and contain images, voices, and videos that require higher network bandwidth. In many cases, corporate firewalls are setup to prevent users within a secured intranet from accessing external sites that may require bi-directional information exchange. This restricts the availability of the web-based service. To avoid those issues of network access, we use e-mails as the media to communicate between the users and the RemoteLab. The e-mail has the advantage that it is widely available and it can be effective with only minimal network resource. In addition, since we are processing the incoming and outgoing mails through a dedicated code, there is little chance for the e-mail based viruses to infect other laboratory computers.

Since the RemoteLab is developed as a network service, the users can submit the job requests as e-mail attachments to a dedicated e-mail account. The students can modify the design template downloaded from specific websites to incorporate the desired experiment content and submit their job requests. Once the e-mails are received by a specific mail server, the experimental server can then access the specific e-mails, isolate the attachments, test for integrity and compile the experiment for execution, upload the binary code to the target processor, run the experiment and record all the specified signals, and send the data back to the students via e-mail. By properly arranging those jobs, the experiment server can accept numerous submissions simultaneously, and the students can receive their results within an estimable time. This enables the limited resource to be available for many users. The design and implementation issues will be discussed in the subsequent sections of this paper.

0-7803-8335-4/04/$17.00 ©2004 AACC

Since the existing infrastructure is used for network access, the RemoteLab does not require any advanced computation resource. In our example, the computers used are the machines retired from computer laboratories. The physical location of the actual experiment setup is no longer a factor. From the users' point of view, the setup can be anywhere. If an institution has a system that adheres to the same protocol, students from other institutions can also access the device. In this framework, institutions do not have to spend the valuable resource to duplicate lab setups when partnering institutions have the same resource. From the lab coordinator's point of view, one can leverage material and experimental designs to provide added diversity in the classroom.

The RemoteLab has been tested for two semesters in a dual level digital control course (ME578) in the School of Mechanical Engineering at Purdue University. During the spring semester in 2003, the system was successfully ported to University of Michigan for testing. Also in the spring semester of 2003, the system was used in distance learning when ME578 was offered through Purdue Continuing Engineering Education program. 9 off-campus students from 5 different sites tested the system. In this paper, we will discuss the results of implementing as well as summarize the architecture and operation of the RemoteLab.

The remaining of the paper is organized as follows. Section II describes the architecture of RemoteLab and the associated hardware and software. Section III states the lessons learned from operating the RemoteLab. The results of applying the RemoteLab in both institutions are discussed in Section IV. Conclusions are summarized in section V.

II. REMOTELAB IMPLEMENTATION

The major tasks of the RemoteLab include: 1) compiling, uploading and running the real-time experiments; 2) job management; 3) communication with external network; and 4) system maintenance and security service capability. The entire system consists of three sub-systems to collaboratively achieve the four major tasks.

A. *Modular System Architecture*

Majority of Internet-based laboratories uses a single server based architecture, where the server is responsible for all aspects of the experiments. The major drawback of this architecture is that once the server is down due to malfunctions or maintenance, and it fails to respond to the submissions. To overcome this shortcoming, the server portion of the RemoteLab is broken into three sub-systems. This architecture, which combines various sub-systems with specific functions, is borrowed from the studies done in the telerobotics area [13].

Fig. 1 shows the architecture of the RemoteLab. The system includes an experimental setup, an experiment server, and a mail server. One advantage of the modular architecture is that the hardware can be chosen to optimize the usage of available resources. The computer used for the experimental setup does not need to be the most up-to-date machine. Those PCs retired from offices or computer laboratories are capable of fulfilling the requirement of laboratory level experiments. On the other hand, the performance of the experiment server limits the through put of the system, and it needs to be implemented on a faster computer with larger memory and disk space. The mail server is the main communication gateway to the external world, and it can be an existing mail server. The experiment server acts as the main conduit for processing experiment jobs. It puts incoming jobs into a queue that has specific priority structure. It also checks and compiles the jobs into binary codes and uploads the codes to the experimental setup. Fault messages and experimental results are then mailed back to the users by the mail server. The experiment server also provides functionalities for lab coordinators to perform maintenance tasks to monitor the status and of the system. The experimental setup simply runs the binary code passed down from the experiment server.

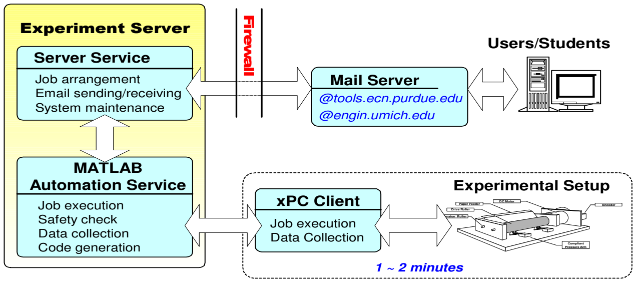


Fig. 1. Architecture of the RemoteLab

One other benefit of the modular approach is that the system is highly flexible. All the three individual sub-systems can be replaced or upgraded anytime without affecting the system functionality. Actually, the experimental setup and the experiment server at Purdue were switched several times throughout the semester without affecting any student tasks.

1) *The Experimental setup*

The experimental setup includes a set of mechanical system and a stand-alone PC that serves as the controller. In the current realization, the real-time kernel supplied with the xPC Toolbox that comes with MATLAB R13 is used. It should be noted that any real-time kernel or any embedded development system can be incorporated into the RemoteLab. The only requirements are that the compiler and linker exist in the experiment server platform and that necessary scripts of MATLAB are developed to compile the Simulink model and to upload the binary codes to the target system.

At Purdue University, the experimental setup uses a retired Pentium Pro 200 computer with 196 MB memory, and the physical hardware of the laboratory setup, as shown in Fig. 2, is an experimental media advance mechanism of an inkjet printer consisting of a DC motor mounted with two helical pinions as the actuator [14]. The main challenge of this system is the nonlinear and position dependent friction in the transmission. At University of Michigan, the experimental setup uses an AMD 333 MHz machine with 128 MB memory salvaged from the junk pile. The mechanism used is a flexible shaft, which is a DC-motor driven system with two heavy flywheels and a slender rod connecting the two wheels (see Fig. 3). The main challenge of this control problem arises from the lightly damped oscillation mode and the temperature-dependent Coulomb friction of the wheel bearings. Note that the hardware setups at Purdue and at Michigan are different, but the RemoteLab service can easily incorporate the different systems and provide a uniform interface for users to interact with the different setups.

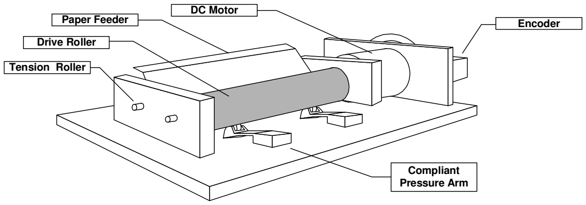


Fig. 2. The experimental setup at Purdue University.

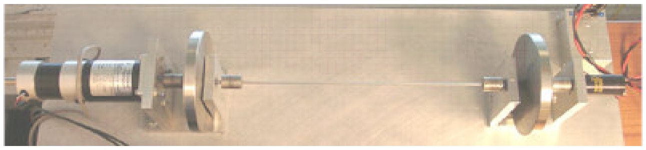


Fig. 3. Flexible wheel experimental setup at University of Michigan.

2) *The Mail Server*

The mail server can be any one that supports POP3 protocol. The server (`tools.ecn.purdue.edu`) used at Purdue University is an existing service at the School of Mechanical Engineering. At University of Michigan, The mail server (`srvr5.engin.umich.edu`) is the College of Engineering CAEN POP3 email server.

3) *The Experiment Server*

The experiment server acts as a service conduit between the experimental setup and the mail server and it runs two main programs: an automation service of MATLAB and a server program developed by *Mechatronics System Research Laboratory* at Purdue University. The server program is developed using Borland C++ Builder and complied for Windows based machines with MATLAB installed. It first derives the submitted jobs from the mail server and arranges the jobs in a priority based job queue. Different priority rules are enforced to form the queue. Only the jobs submitted by valid users will be processed. Once the queue is formed, the experiment server downloads the corresponding e-mails from the mail server, compiles the attached Simulink model (`.mdl`) files into binary codes, and uploads the codes to the target experimental setup with a specific IP address. The server can be rebuilt on any machine by specifying the IP address of the experimental setup and the mail server.

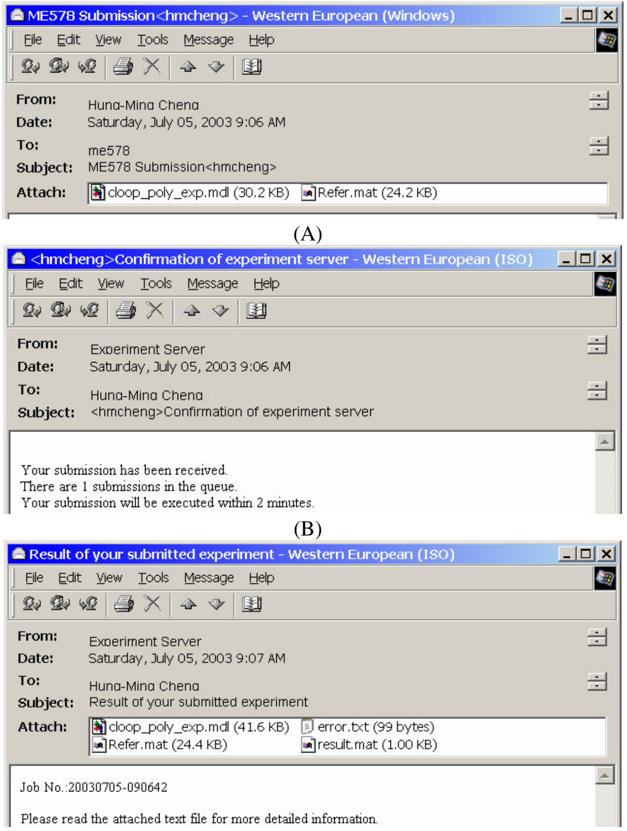
The computation efficiency of the experiment server has a major impact to the response time of the RemoteLab. The experiment server needs to be able to sort, to compile, to link and to upload codes within the experiment time limit, and the real-time implementation time limit is a hard constraint of its capacity. At this point, a 300 MHz Pentium II box is served as the experiment server at Purdue and a 350 MHz Pentium II machine is used at University of Michigan.

B. *Software Interfaces*

The students interact with the RemoteLab through two different software interfaces, the Simulink I/O interface to access the hardware and the e-mail client to submit their jobs.

1) *Simulink I/O Interface*

The students need to have some basic understanding of the hardware mechanism to interact with the RemoteLab. Some related information and the necessary system parameters are available in the lab notes. Two Simulink models are also provided to them to facilitate building the experiments. One of the models is the mathematical model representing the experimental setup. This model is for the students to use as the simulation model to validate their designs before running actual experiments. The other model is an I/O template for accessing the experimental hardware. By examining these two models, students should be able to develop their own experiments to interact with the RemoteLab.


ME578 Submission<hmcheng> - Western European (Windows)
File Edit View Tools Message Help

From: Hung-Ming Cheng
Date: Saturday, July 05, 2003 9:06 AM
To: me578
Subject: ME578 Submission<hmcheng>
Attach: cloop_poly_exp.mdl (30.2 KB) Refer.mat (24.2 KB)

(A)

<hmcheng>Confirmation of experiment server - Western European (ISO)
File Edit View Tools Message Help

From: Experiment Server
Date: Saturday, July 05, 2003 9:06 AM
To: Hung-Ming Cheng
Subject: <hmcheng>Confirmation of experiment server

Your submission has been received.
There are 1 submissions in the queue.
Your submission will be executed within 2 minutes.

(B)

Result of your submitted experiment - Western European (ISO)
File Edit View Tools Message Help

From: Experiment Server
Date: Saturday, July 05, 2003 9:07 AM
To: Hung-Ming Cheng
Subject: Result of your submitted experiment
Attach: cloop_poly_exp.mdl (41.6 KB) error.txt (99 bytes) Refer.mat (24.4 KB) result.mat (1.00 KB)

Job No.:20030705-090642

Please read the attached text file for more detailed information.

(C)

Fig. 4. The example of the submission and the response of the RemoteLab: (A) A submission from the student; (B) the acknowledgement sent by the experiment server; and (C) the results of the submitted job.

2) *E-Mail Interface*

To submit the designed controllers, the only necessary interface that the users need is an e-mail client application.

The application can be any popular e-mail program or web based service. Fig. 4 demonstrates an example of a typical submission process using Outlook Express. To submit a valid job, the subject portion of the mail needs to include specific tags, such as a valid username and the type of job.

After receiving the work, the experiment server sends a confirmation mail back to the address that sends out the submission. The mail acknowledges the recipient of the job request with an estimated completion time based on the number of jobs in the queue. If there is no job waiting in the queue, a job submission should be completed less than 2 minutes depending on the speed of the server. This includes the time of processing e-mail submissions, code compilation, experiment execution, and the processing of the results. After the experiment is done, the RemoteLab sends back the results in MATLAB data file format (`.mat`) as an e-mail attachment, and other information generated, such as job number, system response, error message, and warnings, is also enclosed.

C. *Advantages of the RemoteLab Service*

A traditional laboratory needs instructors attending the lab at specific periods of time and a space for storing numerous setups. The RemoteLab improves the utilization of required space, instrumentations, and man-hour. Through proper job management, the students can share a single laboratory setup and perform the experiments anytime and anywhere without being limited by staffing constraints. In 2003, the RemoteLab used only one setup to serve 21 on-campus and off-campus students in the spring semester at Purdue. The server provides an on-line diagnostic interface and sends out log file to the lab coordinator. The coordinator can regularly track the operation status of the system. This not only guarantees that the students submitting tasks receive the results within a reasonable duration, but also makes the maintenance easier.

Security is one of the major issues on this type of network services. The modulus structure not only provides a better management of network tasks, but also reduces the risks to minimum. The experiment server does not offer any service that allows the users to logon over the network and can be installed on any machine with a dynamic IP address. The only security issue that needs to be taken care of is on the side of the mail server provided by most universities, which should be relatively secure. Since the experiment server does not execute any unrecognized attachments, it is not susceptible to e-mail based viruses. E-mails that do not meet the criterion are truncated immediately and the risks of infecting and propagating e-mail based worms are minimal.

III. Lessons Learned

Implementing the RemoteLab service at Purdue and University of Michigan revealed several issues that are crucial for successful deployment of any remote laboratory realization for on-campus or distance learning.

A. *Software and Hardware Compatibility*

The compatibility of software packages among different versions was one of the biggest issues while implementing the RemoteLab. The software package used to develop the model for the experiment is MATLAB. The design template can be used under either Windows or UNIX environment for the same version of MATLAB. However, the Simulink models are not compatible among different versions. Though the version of MATLAB available on-campus is the most up-to-date one, various versions were used by off-campus students. The links in the model might be missed due to the conversion if the versions are different. The incompatibilities between different versions of the same software are difficult to anticipate. The only way to circumvent this is through developing codes for different versions and rigorous testing. The authors would like to urge software vendors to consider the effort in developing education tools, and provide more downward compatibility or paths to avoid incompatibility between different versions of the same software.

B. *System Robustness*

As mentioned in the previous section, the RemoteLab includes three sub-systems. Individual failure of the three sub-systems causes the entire system to crash. The system might fail to receive e-mails, to upload the compiled code, or to respond to the submission correctly. The robustness discussed in this section includes two parts: the physical setup and the server program. Basically, the server program is as robust as expected. The RemoteLab was maintained once a week. While under maintenance, the experiment server and the experimental setup were rebooted, and all the redundant and temporary files created during the operation were removed. No major crashes were experienced, but several minor failures were reported during the semester.

In 2002, over 3000 jobs were executed, and 4600 submissions were recorded in 2003. The physical mechanism of the experimental setup inevitably wears through usage. Although some degradation is expected, major changes in response pattern needs to be avoided. The physical hardware was validated two to three times each semester to ensure that the system response is within an acceptable range, but some of mechanical failures, such as loosen screws and dropped wires, are unavoidable. These kinds of failures cannot be detected through distance diagnostics. Regular maintenance is required. As a successful laboratory, an early warning mechanism and a detailed maintenance guide are necessary.

The RemoteLab was set to check for new submissions every 5 minutes, and all operations were recorded to a log file. Connection timeout is the most common failure, and the two main reasons of this error are unexpected maintenance shutdown of the mail server and network congestion. It is thus very important that better error handling is required. Close collaboration with the mail server is also necessary.

Especially, the significant mail traffic always causes concerns of network administrators before the due dates of homework and final project.

C. System Security

One of the major concerns of the RemoteLab is security. Although mail-based viruses are not an issue, unfriendly intruders have been encountered. Several unauthorized trials to gain access to the experiment server were detected and blocked by a software firewall installed in 2002. Although none of the attempts was successful due the existence of the firewall, the numbers of these hacking attempts is alarming. Since the server program is developed for Windows based systems, it is important to apply all relevant security patches and a firewall system is necessary. The experiment server was protected by a hardware firewall in 2003 and no security breaches were detected. Another issue is unauthorized submissions. By design, the system is open to all the people who have access to the Internet. This implies that any job with the proper access protocol will be executed. In 2003, an authorized-user checking mechanism was implemented. In addition to manage proper access, it also provides the lab coordinator a method to monitor student activity and to initiate preventive measure.

D. Safety

The most important issue at all, the RemoteLab must be safe. Not only do people and property in the lab need to be protected from the operation of the unanticipated experiment, but also in the case of any mishap the remote user will not be there to intervene. The RemoteLab must be able to tolerate a range of errors and still be a safe environment. Especially, users who first use the system might make mistakes. Several automatic safety-checking mechanisms are enabled to ensure that students and the equipment are protected. Once the experiment fails to meet the criterions enforced in the server program, such as power saturation for preventing the over current and forced-stop for corrupted code, the RemoteLab will not run the job and an error message is recorded a log file. The log file is used to record all the accesses and operations of the RemoteLab and sent to the lab coordinator daily. The purpose of the log file is to trace the system failure and to respond to the questions from students. Since students have 24/7 access to the RemoteLab, the lab coordinator needs to be able to monitor the operation both on-site and through a secure remote desktop and perform necessary maintenance and remotely respond to issues and failures. The system can be turned off or reset remotely if any critical situation occurs.

IV. Results

The RemoteLab has been tested in the spring semesters of 2002 and 2003 at Purdue and in the spring semester of 2003 in a graduate level digital control course at University of Michigan. The Purdue students included both on-campus and off-campus students. The off-campus students came from 3 different states and 5 different companies. The students at University of Michigan were all on-campus students.

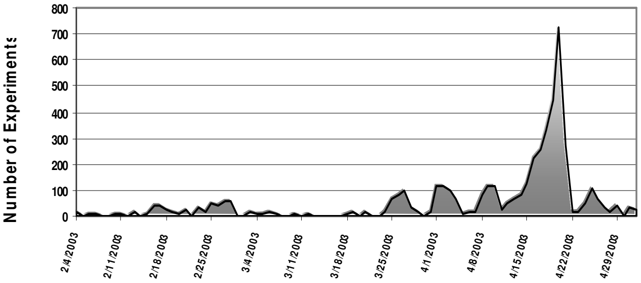


Fig. 5. Distribution of the experiment jobs during the semester.

RemoteLab enabled the students to submit their designs at home or in their offices. The daily usage of the system at Purdue during the 2003 spring semester is shown in Fig. 5. The traffic peak occurred during the week before the deadline of the final project. There were more than 720 submissions to the experiment server within a single day. To maintain equal access, the following rules were enforced: 1) each student is allowed to submit only one job at a time; 2) all jobs are limited to 10 seconds; and 3) all submitted jobs must be simulated. The first two rules are coded in the server program. All the jobs violating the rules are truncated with a warning mail sending back to the user.

Similar usage patterns were observed from University of Michigan students. The RemoteLab ran quite successfully, with more than 2500 jobs successfully served through the semester. The students, in general, reacted quite favorably to the RemoteLab. There is only one peculiar and unexpected problem experienced at University of Michigan. During the semester, when the mail account suddenly experienced a dramatic increase in the number of received e-mails, the received SIMULINK files become scrambled. It seems there is an anti-spam or snooping mechanism installed in the mail system, which caused the file corruption. The corrupted script file will then cause a fatal error during the compilation. This phenomenon always happened right before the deadline of the homework, which is very frustrating for the students. When students were asked to stop sending emails for several days, subsequent job submissions usually became normal.

TABLE I SUMMARY OF THE STUDENT RESPONSES

	(Agree)				(Disagree)	
	5	4	3	2	1	Avg
Inclusion of experiment in homework is helpful	11	4	4	0	1	4.20
Course project is helpful	13	7	0	0	0	4.65
Access other systems	11	6	2	1	0	4.35
Compete with other university	9	5	5	1	0	4.10
Like to participate in similar activities	11	3	4	1	1	4.10

Students' opinions regarding to the RemoteLab were surveyed in the end of the semester. Table I lists the number of students' responses to the survey. Most of the students agree that the experiments and the final project are helpful for the class. Table II lists the percentage of the students' responses to the RemoteLab. Not surprisingly, convenience is listed by the highest percentage of students. Students also agree that the system is hands-on and easy to implement controllers. From the result of the feedback, they were highly interested in implementation and practical experience. They also voiced their opinions about how to improve the situation, such as an automatic warning mechanism for maintenance and system backup schedule. A lot of students asked for more work on the experiments for the class, and they would like to have more options to try out different physical systems.

TABLE II STUDENTS' PERSPECTIVE OF THE REMOTELAB

	(Agree) 5	4	3	2	(Disagree) 1
RemoteLab improved my learning?	50%	50%	0%	0%	0%
Convenient?	84.21%	15.79%	0%	0%	0%
Almost hands-on experience?	38.89%	44.44%	16.67%	0%	0%
Easy to implement controllers?	75%	25%	0%	0%	0%

Several shortcomings were also mentioned in the feedback. Since all the wires and sensors are well connected by the lab coordinator, students who took the course did not need to debug or check the connection of laboratory setup. Students might not have any feeling without practical operations on the physical experiments. Therefore, although the actual system is on-campus, some of the students still treated the experiment as a simulation process. In some web-based remote laboratories, the video conferencing scheme is utilized [4][5], which feeds the video stream and the audio effects back to the students and makes students to have the feelings of performing a real experiment.

Furthermore, though the rules of job submission were specified in the semester in 2002, they were not enforced strictly. Some students tended to dump a huge amount of jobs to the system before the homework due. This not only abused the system, but also led the queue jammed. All other users needed to wait till all the jobs in the queue being executed. The worst case took around 2 hours for a piled queue during the semester. The one-job-per-time policy was enforced strictly and coded in the server program in 2003. However, students still tended to dump a huge amount of jobs to the server. The most extremely case was that there were more than 300 jobs waiting in the queue, and those jobs took more than four hours before the server can function normally.

V. CONCLUSION

The RemoteLab has been found to be a cost-effective means for conducting collaborative work through the sharing of unique resources. The effective implementation of the remote lab takes time, especially in the beginning stage. Successful integration of the Internet-based tools and the traditional engineering lab within the course opens up unprecedented possibilities for learning, communication, information exchange, and interactivity.

Assuming that enough effort can be put into the creation of interactively engaging on-line resource, one would predict that eventually the Internet-based and self-paced mode of instruction will become more and more important in the future. The current implementation provides an excellent base upon which additional capabilities can be built. The next step in this development will be making the system a more reliable and easier to maintain environment.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the assistance of W. H. Wiegman, P. M. Roales, and D. Howard for helping organizing the work. The students who have taken Digital Control over the past two years are also gratefully acknowledged for their pioneering spirit and valuable feedback.

REFERENCES

[1] C. Williamson, J. T. Bernhard, and K. Chamberlin, "Perspectives on an Internet-Based Synchronous Distance Learning Experience," *J. Engineering Education*, vol. 89, no. 1, pp. 53-62, January 2000.

[2] K. G. Paterson, "Student Perceptions of Internet-Based Learning Tools in Environmental Engineering Education," *J. Engineering Education*, vol. 88, no. 3, pp. 295-304, July 1999.

[3] B. Aktan, C. A. Bohus, L. A. Crowl, and M. H. Shor, "Distance Learning Applied to Control Engineering Laboratories," *IEEE Trans. Education*, vol. 39, no. 3, pp. 320-326, Aug 1996.

[4] C. C. Ko, B. M. Chen, J. Chen, Y. Zhuang, and K. C. Tan, "Development of a Web-Based Laboratory for Control Experiments on a Coupled Tank Apparatus," *IEEE Trans. Education*, vol. 44, no. 1, pp. 76-86, Jan 2001.

[5] H. Shen, Z. Xu, B. Dalager, V. Kristiansen, Ø. Strøm, M. S. Shur, T. A. Fjeldy, J. Q. Lű, and T. Ytterdal, "Conducting Laboratory Experiments over the Internet," *IEEE Trans. Education*, vol. 42, no. 3, pp. 180-185, Aug 1999.

[6] M. Shor and A. Bhandari, "Access to an Instructional Control Laboratory Experiment through the World Wide Web," in *Proc. 1998 Amer. Contr. Conf.*, Philadelphia, PA, 1998, pp. 1319-1325.

[7] E. Guimaraes, A. Maffeis, J. Pereira, B. Russo, E. Cardozo, M. Bergerman, and M. F. Magalhaes, "REAL: A Virtual Laboratory for Mobile Robot Experiments," *IEEE Trans. Education*, vol. 46, no. 1, pp. 37-42, February 2003.

[8] W. R. Ferrell, and T. B. Sheridan, "Supervisory Control of Remote Manipulation," *IEEE Spectrum*, pp. 81-88, October 1967.

[9] G. V. Kondraske, R. A. Volz, D. H. Johnson, D. Tesar, J. C. Trinkle, and C. R. Price, "Network-Based Infrastructure for Distributed Remote Operations and Robotics Research," *IEEE Trans. Robotics and Automation*, vol. 9, no. 5, pp. 702-704, October 1993.

[10] J. W. Overstreet and A. Tzes, "An Internet-Based Real-Time Control Engineering Laboratory," *IEEE Control Systems Magazine*, vol. 19, no. 5, pp. 19-34, October 1999.

[11] *Remote Dynamical System Laboratory*. Available: http://dynamics.soe.stevens-tech.edu.

[12] *Ask a Microscopist!* Available: http://www.msa.microscopy.com/Ask-A-Microscopist.html.

[13] M. W. Gertz, D. B. Stewart, and P. K. Khosla, "A Human Interface for Distributed Virtual Laboratories," *IEEE Robotics and Automation Magazine*, vol. 1, no. 4, pp. 5-13, December 1994.

[14] C.Y. Chen, C.C. Cheng, and G. T. -C. Chiu, "Adaptive Robust Control of Media Advance Systems for Thermal InkJet Printers," *International Journal of Mechatronics*, vol. 10, iss. 1-2, pp. 111 – 126, 2000.