01 Aug 2007

A Chemical Engineering Laboratory over Distributed Control and Measurement Systems

Jing Guo

David Kettler

Muthanna H. Al-Dahhan
Missouri University of Science and Technology, aldahhanm@mst.edu

Recommended Citation

J. Guo et al., "A Chemical Engineering Laboratory over Distributed Control and Measurement Systems," *Computer Applications in Engineering Education*, vol. 15, no. 2, pp. 174 - 184, Wiley, Aug 2007.
The definitive version is available at https://doi.org/10.1002/cae.20108

The Digital Engineering Symposium

40+ sessions covering multiple MBSE platforms over four days in four tracks, giving you novel solutions to real digital engineering, systems engineering, and enterprise architecture problems.

KEYNOTE PRESENTATION

Dan Thurmon

Founder and President of Motivation Works, Inc, helping leaders and organizations navigate change.

For the Designers

Drive intentional outcomes by weaving stakeholder needs into engineering requirements. For early-lifecycle professionals.

For the Makers

Realize ideas by connecting upstream and downstream engineering processes. For design, verification, and operations.

For the Thinkers

Solve complex problems that impact timelines and budgets regardless of lifecycle stage. Learn from industry thought-leaders.

For the Innovators

Breaking boundaries - Meet the mavericks that are paving the way for digital engineering's future.

+ One registration,
Two events

ZUKEN innovation WORLD

Attend Zuken Innovation World sessions - Included with your Integrate23 registration .

More Info. | Register

Hosted by

Vitech
A Zuken Company

vitechcorp.com

A Chemical Engineering Laboratory Over Distributed Control and Measurement Systems

JING GUO, DAVID KETTLER, MUTHANNA AL-DAHHAN

Department of Energy, Environmental and Chemical Engineering, Chemical Reaction Engineering Laboratory (CREL), *Campus Box 1198, Washington University, St. Louis, Missouri 63130*

Received 5 August 2005; accepted 2 August 2006

ABSTRACT: Remote control of lab apparatus via the Internet allows flexible scheduling of experiments and exercises. This paper presents a novel application of computers and Internet technology in chemical engineering education that highlights interactive learning for an undergraduate course in online measurement and control. The proposed client–server architecture can be adapted to drive different kinds of devices and can be easily expanded. The developed online laboratory has been tested by many students using different connections and operating conditions. This Internet-based solution can improve undergraduate engineering education in a conventional unit operation laboratory. As an example, the application of remote experiments to characterize a plug flow reactor is described. © 2007 Wiley Periodicals, Inc. Comput Appl Eng Educ 15: 174–184, 2007; Published online in Wiley InterScience (www.interscience.wiley.com); DOI 10.1002/cae.20108

Keywords: Internet-based educational laboratory; online control; remote measurement; server–client application

INTRODUCTION

Laboratory exercises have been an essential part of many university educational programs. Local labs are the best way to get hands-on experience in operating laboratory devices. In a traditional university laboratory, students work in teams with tutorial help from teachers. However, because the Internet technologies are becoming increasingly important in engineering education, new types of online laboratories based on computers and the Internet are being created. Combining the physical setup of local labs with the flexibility of remote access, online laboratories offer remote access to laboratory equipment, workbenches, and many types of experiments. The web-based system overcomes many obstacles in performing a unit operation laboratory in its current setting: space and time limitation, experiment-related hazards and safety concerns, and decreasing resources for

Correspondence to J. Guo (jingguo@che.wustl.edu).

performing actual experiments [1]. As well as the convenient learning environment, economy and cost issues are extremely important. Online laboratories hold promise of being up to two orders of magnitude cheaper than conventional ones [2].

A real-time, Internet mediated-laboratory brings several advantages to the university. First, it provides schedule flexibility and access to more students. Because the real-time experiment can be run anywhere at any time across the Internet, it may be assigned as homework, without the need for scheduling physical meetings between students and instructors. To run the experiment remotely, even from home, students consult a web page or downloadable file for instructions. Then they download a client application program and connect to a specified server. The students and instructors can spend less time on communicating the operating procedure and more time on researching the experimental results. With the connectivity of the Internet and new generations of software applications, remote learning has evolved into a new model, which provides higher quality and more flexibility in an anytime/anywhere fashion. This new approach is not to provide more information, but to engage students in a unique mode of learning that builds a solid foundation, while providing good examples of learning techniques [3].

A second advantage of online laboratory is that it increases educational quality. Rather than several universities spending money on the same equipment for the same experiments, cooperating universities may each carry out one unique experiment and then form an experiment pool. A rich online resource for undergraduate chemical engineering students doing process design subjects can be provided [4]. Universities can put their experiments on the Internet, where the content can be easily updated through electronic document publishing. Thus universities, departments, or individual instructors can maintain and execute experiments in laboratory environments that would otherwise be too costly, too time consuming, or too difficult to maintain and execute individually.

Finally, online laboratory brings increased recognition and strength to the originating university. Because the experiment may be run anytime and anywhere, the location of the university will not be a factor in a prospective student's decision to register at a school, only the quality of the education it offers. Students from around the world will be able to register and take courses from the university with the best offering of online experiments and classrooms. The delivery of such a system over the world-wide-web is a logical choice because of increasing accessibility bandwidth. In addition, advances in the Internet and emerging modeling tools make it easier and more cost effective to develop web-based educational or laboratory systems. The only requirement is maintenance of a single web site on a reliable server; there is no need to worry about client sites [5].

The web has rapidly imposed itself as a new medium for interconnecting people throughout the world. It has been used in a variety of applications to provide platform-independent access to relevant information. However, combining heterogeneous tools with applications remains an important and only partially solved problem. Additional programming is required in order to create interactive web-based forms and applications [6]. In this work, we implemented an interactive online laboratory system over the Internet to be used from a remote location as part of a distance learning strategy. A variety of tools and programs are fully integrated and run as a single application. Our research designed a framework for chemical engineering education which brings together a user's interests and active participation in the lab. Then its effectiveness was evaluated through experimentation with undergraduate students taking the unit operation lab course. This work presents the tracer experiment as one example to illustrate the manipulation of instruments and real devices for web-based education in automatic control and measurement.

EXPERIMENTAL SETUP

A tracer study experiment is carried out in a tubular reactor in the Chemical Engineering Laboratory at Washington University. The non-ideality of the reactor performance is evaluated using a tracer. After a pulse of tracer is inserted with the feed to the reactor, influent and effluent tracer concentrations are monitored via the solution conductivity measurement. The recorded data are used to construct a response curve in order to reveal important characteristics of the reactor flow, such as dead zone volume, dispersion, recirculation, and channeling. All these deviations from ideality must be accounted for to accurately calculate the conversion, reaction time, and necessary reactor size for a given process. The experiment is run both locally in the laboratory and remotely in real-time over the Internet.

Figure 1 displays the setup built for the tracer study experiment. Starting at the point labeled ‘‘WATER SOURCE,’’ a clear plastic hose with a ball valve attached to the end connects a header containing

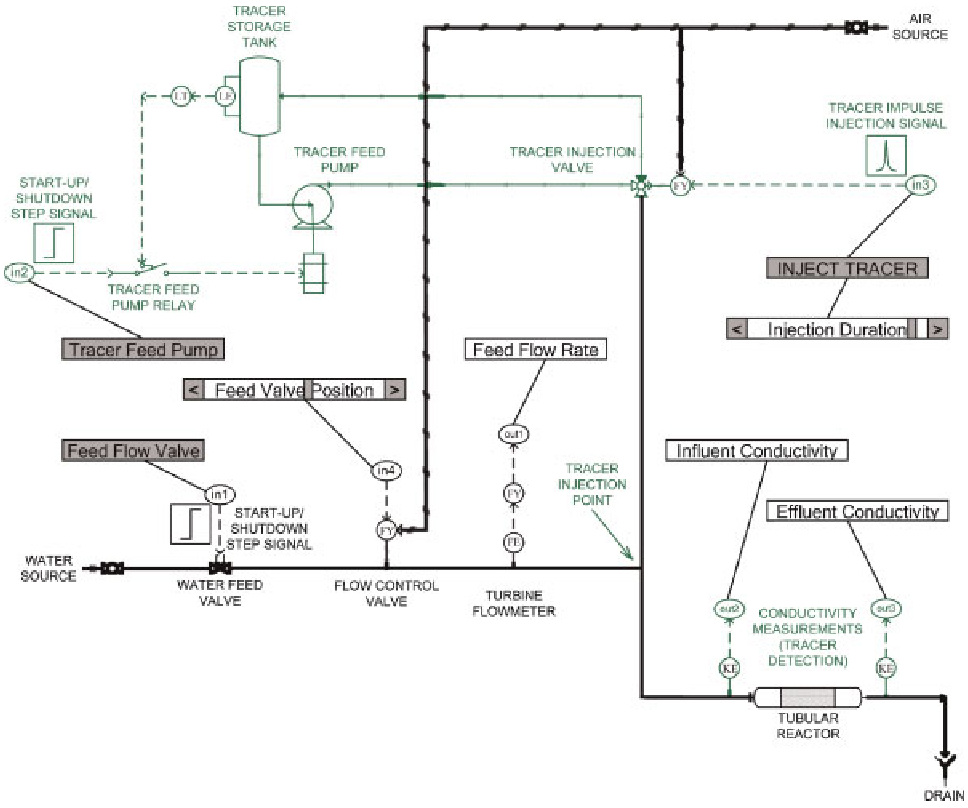


Figure 1 Flow diagram of tracer study experiment process. The dashed lines represent electrical measurements or control signals. The lines with x marks represent the tracer loop. Pneumatic lines are also marked with double slashes for those control elements with air pressure as their driving force. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

the laboratory's water source to the process's water feed pipe via a nozzle. The water fed to the pipe is released downstream by the solenoid valve when the student pushes the ''Feed Flow Valve'' button on the downloaded client application. The student controls the position of an air-actuated, air-to-open, feed flow control valve located further downstream with a ''Feed Valve Position'' scroll bar on the client application. The ''Feed Flow Rate'' is measured by a turbine flow meter located just before the tracer injection point, and the measurement is sent to the client application. From there the feed enters the tubular reactor via a plastic tube and exits the reactor to the drain. Two conductivity sensors, one at the reactor influent and one at the reactor effluent, measure the ''Influent Conductivity'' and ''Effluent Conductivity'' of the reactor feed and product streams, respectively, and send this data to the client application.

A plastic storage tank holds KCl solution with a concentration of 1 g/L. A level detector senses the amount of tracer solution in the tank; when at an adequate level, the signal from the level detector closes a relay switch which allows the tracer feed pump to circulate the tracer solution from the storage tank to the injection valve and back to the storage tank via plastic tubing. The tracer pump is activated when the student pushes the ''Tracer Feed Pump'' button on the client application. Once the tracer solution is circulating, the student uses the ''Injection Duration'' scroll bar to set the duration that the injection valve will remain open. The student activates the injection valve by pushing the ''INJECT TRACER'' button on the client application. The injection valve is an air actuated air-to-open solenoid valve. A plastic hose with a ball valve attached to the end connects the laboratory's air source header to the process's air feed tubes via a nozzle. The feed flow control valve requires 40 psi of air, while the tracer injection valve requires 100 psi of air. When the tracer is injected, a finite volume of tracer solution enters the feed tubing exiting the turbine flow meter, where its conductivity is measured upon entering and exiting the tubular reactor.

DEVELOPMENT OF SYSTEM ARCHITECTURE

Design of Server–Client Architecture

The research efforts developed various simulators, related software libraries, laboratory control schemes, and local measuring devices. The necessary information regarding the lab experiment, including theoretical background and user manuals, is provided on web pages. Users can actively learn through interaction with the system, at any time and place over the Internet. The Internet offers a number of protocols. The hyper text transfer protocol (HTTP) that forms the basis of the web uses uniform resource locators (URLs) to locate data on the Internet. The connection between the server and client program is made using a TCP/IP Winsock socket located within both programs, which functions much like a phone receiver/dialer on each side of the Internet. Figure 2 shows a signal flow chart for remote automatic control of the lab experiment. To control the process, the client builds a TCP/IP socket connection to an I/O server that sends the output signals to the controller, reads the input signals, and sends them back to the client. The client program's TCP/IP socket connects to the server program's TCP/IP socket by using the server's IP address on the Washington University network. In order to make the safe connection, the same local port number must be specified within both the client and server TCP/IP Winsock sockets.

The server sends measurement data to the client the same way the client sends control commands to the server, by creating a string of numbers representing all the commands or measurements and sending them through the TCP/IP socket. After the server receives the control string of numbers, the string is parsed and each measurement or command within the string is sent to its appropriate subroutine within the server code ''Server_TracerStudy.'' Once the client TCP/IP socket connects to the server TCP/IP socket and receives the text in ''OutputString,'' it separates each of the measurement variables in the string and sends them to the textboxes located on the right side of the user interface of the client program ''Client_TracerStudy,'' as shown in Figure 2. It is simple to add a new measurement or control variable. About several lines of code inserted in the server and client code will add more numbers representing the additional measurement or control variable to the string.

The server can watch over all user interactions, so user interactive programming can be more easily achieved. Sequences of operations, direct user inputs, user response times, and information requests are used as input data for the user profile updates. However, these may excessively increase the network traffic if high interactivity is required. Server overload may happen if more clients are trying to connect to the server than the number expected in the design of the system. To authenticate the user information and control the server load, a username and password check is employed.

Design of Server–Experiment Architecture

The interface between the server and the experimental setup is via a data translation card, which can be installed directly in the expansion slots of the server. Table 1 lists all of the process variables used as signals in the tracer experiment. Digital signals are either on or off when equal to 1 or 0, respectively. Analog signals send (Output) or receive (Input) signals within a defined range of values, shown in the ''Range'' column in Table 1. The ''Bit/Interface #'' column corresponds to the number specified in the Labtech runtime program ''Traceexe.ltc'' as the point where the signal wires connect to the interface card in the server computer. Figure 2 also illustrates how the signal wires are connected to the data translation DT9804 interface board. The measurement variables are ''Runtime'' (s), reactor ''Influent Conductivity'' (mS), reactor ''Effluent Conductivity'' (mS), and reactor ''Feed Flow Rate'' (L/min).

System Hardware/Software Setup

The server is a Pentium-IV personal computer running on Windows 2000. The server uses drivers from data translation to access the DT9804 interface board. The server computer connects to the board using a USB cable, and the interface board has analog input/output and digital input/output ports for connection to the physical control hardware of the reactor system. The server is connected to the university network via the Internet. Since Internet-related technologies are changing rapidly, we face many choices of tools and technologies in implementing our proposed system.

Programming in mixed languages is adopted, and Visual Basic (VB) is favored for user interface development. Even though Java applets can provide good control via pop-up windows that provide interaction and communication directly with the server, unsigned applets make it less safe for the client to store the measurement data and transfer them to other applications. A further problem with Java is that the functionality of an applet may vary between different browsers. Therefore, Visual Basic, with its relative

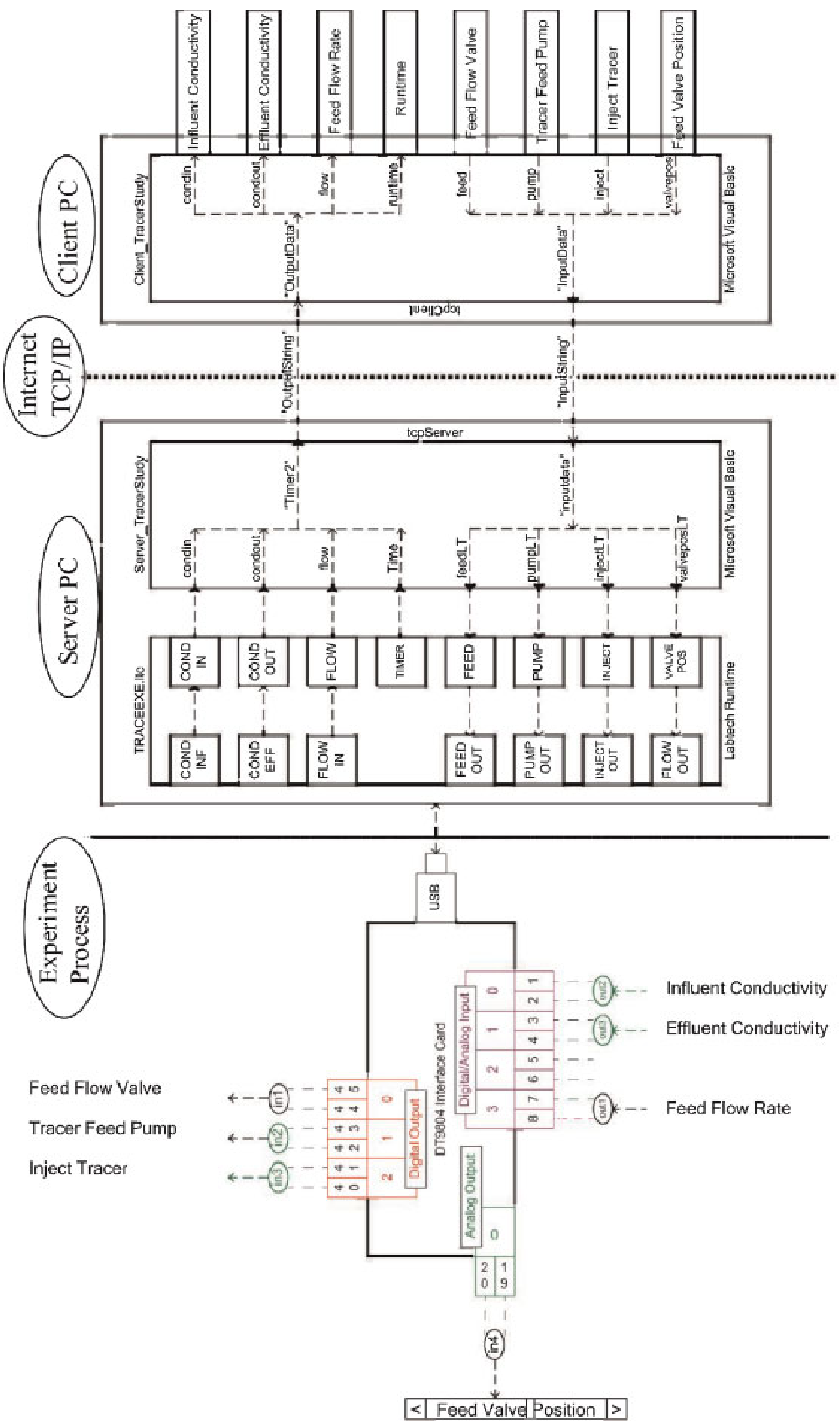


Figure 2 Signal flow chart for remote automatic control of the tracer study experiment. In the client and server computer applications, the signals are combined as a string before being sent over the Internet. Once received, the string is parsed back into individual measurement or control variables. The involved subroutines are within the programs “Server_TracerStudy” and “Client_TracerStudy.” [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Table 1 Process Variables Involved in the Experiment Design

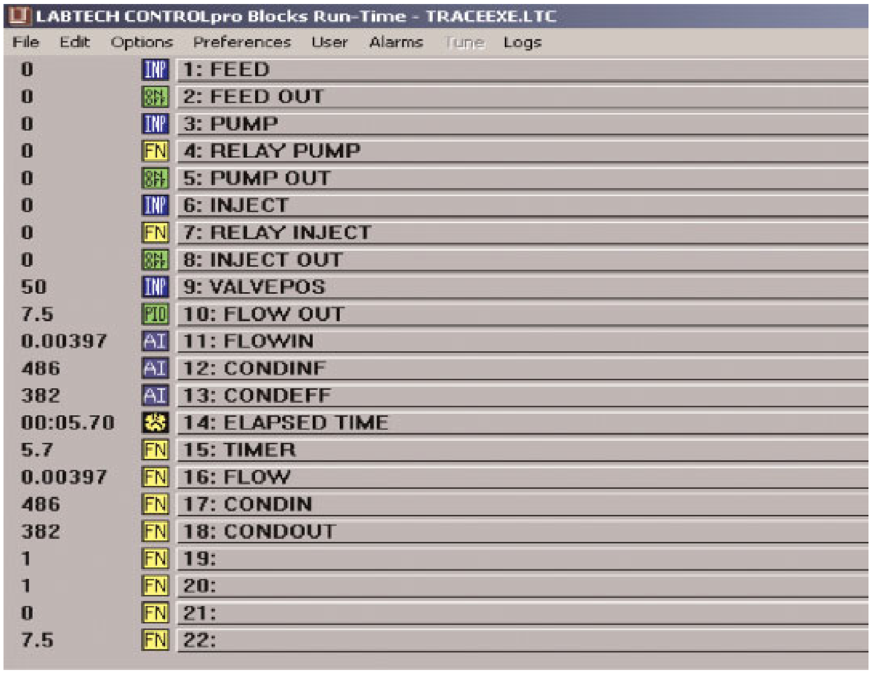
Variable	Range	Units	Type	Origin	Destination	Bit/interface #
Reactor feed valve	0–1	V	Digital output	Control Hardware	‘‘Client_TracerStudy.exe’’	0
Tracer feed pump	0–1	V	Digital output	Control Hardware	‘‘Client_TracerStudy.exe’’	1
Inject tracer	0–1	V	Digital output	Control Hardware	‘‘Client_TracerStudy.exe’’	2
Tracer injection duration	1–3	s	Analog output	‘‘Server_TracerStudy.exe’’	‘‘Client_TracerStudy.exe’’	n/a
Reactor feed valve position	0–100	%	Analog output	Control Hardware	‘‘Client_TracerStudy.exe’’	0
Run time	>0	s	Analog input	‘‘Client_TracerStudy.exe’’	‘‘Traceexe.ltc’’	n/a
Reactor: influent conductivity	>0	mS	Analog input	‘‘Client_TracerStudy.exe’’	Control Hardware	0
Reactor: effluent conductivity	>0	mS	Analog input	‘‘Client_TracerStudy.exe’’	Control Hardware	1
Reactor: feed flow rate	>0	L/min	Analog input	‘‘Client_TracerStudy.exe’’	Control Hardware	3

simplicity and duplicity, was employed to develop the user interface. Other codes were constructed via the Labtech ControlPro 12.1 runtime program, which receives data and sends commands to the control hardware. Labtech can be programmed to control the hardware by drag-and-dropping control icon blocks from its menu into its build-time workspace. The primary blocks include digital output, analog output, analog input, calculation blocks, log blocks and the IPC input block. These blocks are contained in ‘‘TRACEEXE.ltc,’’ which must be started before running the server. Figure 3 illustrates the appearance of ‘‘TRACEEXE.ltc’’ while it is running. The server program ‘‘Server_TracerStudy.exe,’’ designed using Microsoft Visual Basic 6.0, is run subsequently. These two programs must be executed before a client can access the experiment using the client program, ‘‘Client_TracerStudy.exe.’’ Details will be given in the next section.

SYSTEM IMPLEMENTATION

Server Applications

The first application in the server computer to receive the measurement signals from the USB port on



```
LABTECH CONTROLpro Blocks Run-Time - TRACEEXE.LTC
File  Edit  Options  Preferences  User  Alarms  Tune  Logs
0           INP     1: FEED
0           ON/OFF  2: FEED OUT
0           INP     3: PUMP
0           FN      4: RELAY PUMP
0           ON/OFF  5: PUMP OUT
0           INP     6: INJECT
0           FN      7: RELAY INJECT
0           ON/OFF  8: INJECT OUT
50          INP     9: VALVEPOS
7.5         PID     10: FLOW OUT
0.00397     AI      11: FLOWIN
486         AI      12: CONDINF
382         AI      13: CONDEFF
00:05.70            14: ELAPSED TIME
5.7         FN      15: TIMER
0.00397     FN      16: FLOW
486         FN      17: CONDIN
382         FN      18: CONDOUT
1           FN      19:
1           FN      20:
0           FN      21:
7.5         FN      22:
```

Figure 3 Labtech runtime Windows for ‘‘TRACEEXE.ltc.’’ This is the window actually seen when the program is run in conjunction with the server application. The values on the left side are the actual voltage values interfacing with the interface board to the control hardware. The order of the blocks is an indication of the order in which they were created for this program. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

the interface card is the Labtech runtime program “Traceexe.ltc.” All measurement variables are classified as analog inputs in “Traceexe.ltc.” Each analog input block in “Traceexe.ltc” receives the correct signal from the interface card by specifying the correct interface point within its properties. For example, as shown in Figure 2, the “Feed Flow Rate” signal wire is connected to label “3” on the DT9804 interface card, so the interface point in “Traceexe.ltc” for “Feed Flow Rate” has to be specified as “3.” Figure 4 is an illustration of the control blocks in “Traceexe.ltc.” The control block (set to “X”) allows the programmer to set up a Visual Basic-designed server application from Labtech by creating a “GetLT” function in the server application. Once these functions are typed into the Visual Basic server application code, the programmer can add one line of code under the appropriate subroutine within the sever code and specify the corresponding application, topic, and item in the IPC input block for each data stream. The measurement variable, time, comes from the “Timer” in “Traceexe.ltc.”

The second application in the server computer, “Server_TracerStudy.exe,” receives the measurements from “Traceexe.ltc” by continuously using the GetLT function. This function uses a built-in Labtech application called LT-Speedway to grab the analog input data received by “Traceexe.ltc.” The program “Server_TracerStudy.exe” takes four measurement variables it receives from its GetLT function and combines them into one string of text, called “OutputString.” After the client connects to the server computer using the client program “Client_TracerStudy,” “Server_TracerStudy.exe” sends “OutputString” across the Internet once every 100 ms to the client program, using a timer within the server program called “Timer2.”

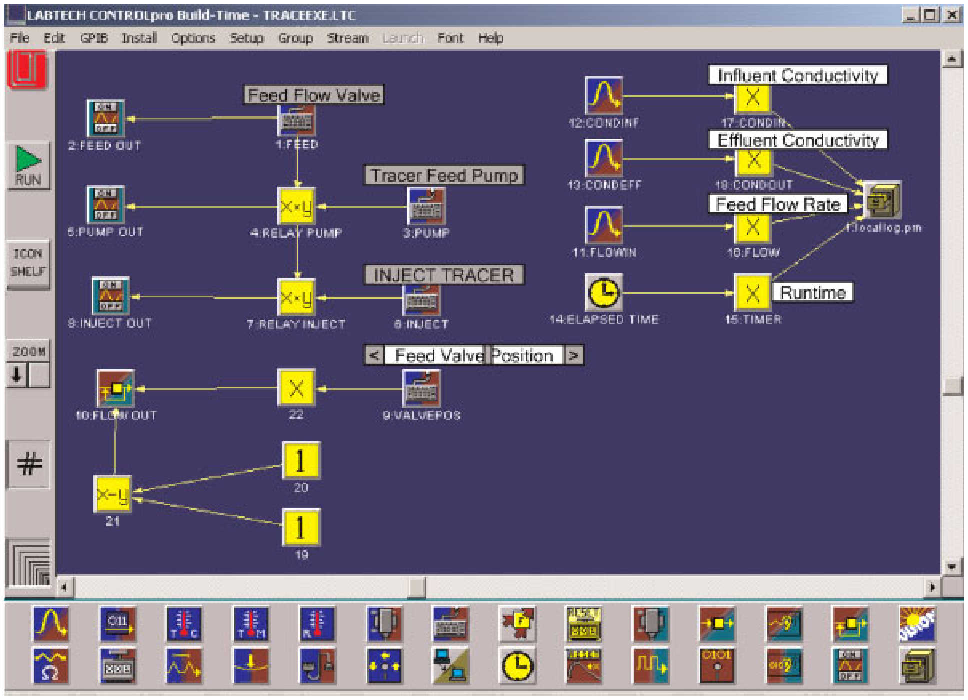
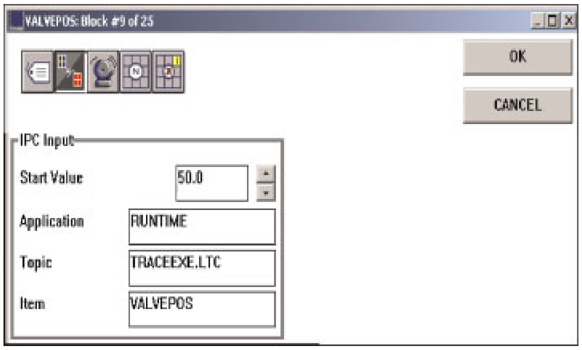


Figure 4 Labtech build-time visualization of the Runtime program “TRACEEXE.ltc.” Icons 1, 3, 6, and 9 are the IPC input blocks which receive control commands from the server application. Icons 2, 5, and 8 are digital outputs for the feed flow valve, tracer pump and tracer injection valve, respectively. The icon 10 is an analog output for the valve position. Icons 11, 12, and 13 are analog input boxes from the conductivity sensors and the flow meter. Icon 14 is the timer which sends the runtime to the server. The icon with the filing cabinet is a log which keeps local records of the measurement data on the local server in a file called “locallog.prn.” The “X × Y” icons are safety relay switches. The “1” and “X – Y” icons create a setpoint of zero for the analog output block. The “X” boxes create a port to send measurements to the server application. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]



VALVEPOS: Block #9 of 25

IPC Input

Start Value 50.0

Application RUNTIME

Topic TRACEEXE.LTC

Item VALVEPOS

Figure 5 Labtech Build-time visualization of the IPC Input Parameters for the IPC Input Block. The Application, Topic, and Item labels are used in the command line that begins with PutLT under the ''InputData'' subroutine in the server application ''Server_TracerStudy.exe.'' [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

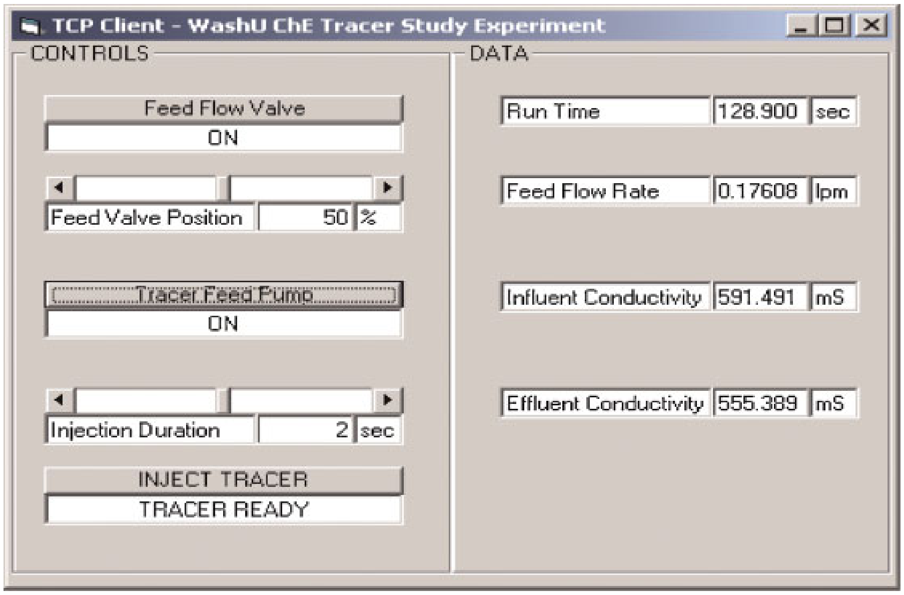
A ''PutLT'' function is created in the IPC input block to communicate with Labtech from a Visual Basic-designed server application. Figure 5 illustrates an example of the input fields in an IPC input block. The ''Application'' is ''runtime,'' and ''Topic'' is the name of the runtime application that the server program is using to communicate with the control hardware. ''Item'' is the name of the control variable used in the PutLT function in the server program. In the server application, the function PutLT takes each of the values in the textboxes on the left-hand side of the server application user interface and sends them to their respective IPC input block in ''Traceexe.ltc.'' The IPC input block receives each signal in ''Traceexe.ltc'' and sends it to its corresponding bit number (digital) or interface point (analog) on the DT9804 interface card. For example, when the client gives the command ''INJECT TRACER,'' ''1'' is put into the correct place within the ''InputString'' and sent to the server application through the client TCP/IP socket. Once the server application receives ''InputString'' from its TCP/IP socket, as shown in Figure 2, it separates the ''1'' from ''InputString'' within its function ''InputData'' and sends it to the ''Inject'' IPC input block in ''Traceexe.ltc'' using its PutLT function. The IPC input block in turn sends ''1'' to the ''Inject Out'' digital output block in ''Traceexe.ltc.'' The digital output, as illustrated in Figure 2, sends 10 V from point #2 on the DT9804 interface card to the tracer injection valve, which implements the client's command of ''INJECT TRACER.''

Client Applications

The aforementioned two server programs should be up running before a client can access the experiment using the client program, ''Client_TracerStudy.exe.'' At first the client downloads this program from a web page and stores it on the remote computer. Once the client double-clicks on the icon, it opens up and connects to ''Server_TracerStudy.exe'' on the server. Every command the client manipulates sends text data from the client TCP/IP socket across the Internet to the server TCP/IP socket. The server program sends measurement data acquired from the Labtech runtime program ''Traceexe.ltc'' back to the client through its TCP/IP socket. It also sends the control variable commands acquired from the client to ''Traceexe.ltc,'' where the commands are executed on the control hardware.

Figure 6 is the user interface of the client program, ''Client_TracerStudy.exe.'' The controlled variables are reactor feed ''Valve Position'' (%open) and tracer ''Injection Duration'' (s). These are analog outputs, which are set by the client using a scroll bar on the user interface before the tracer is injected. The client then switches on the reactor ''Feed Flow Valve,'' the ''Tracer Feed Pump,'' and ''Inject Tracer'' by pushing their respective buttons on the client user interface. These are digital outputs in ''Traceexe.ltc.'' Whenever the client slides one of the scroll bars (analog signals) or pushes one of the buttons (digital signals) on the client user interface, the current values of all the analog and digital outputs are combined as a string of text called ''InputString'' in the InputData function in the client program. ''InputString'' is sent to the server program through the client TCP/IP socket. The server program picks up the ''InputString'' string of text across the Internet at its TCP/IP socket, separates all of the outputs, and places them in their respective textboxes on the left hand side of the server application user interface, shown in Figure 7.

The laboratory session manager can identify the users logged on the system to interact with the required instruments for a given lab session. The manager thus knows the number of users connected to the measurement instrument, each user's access rights to the remote instrument (read-only access or access with property modifications), and the name of the user who has the right to modify the measurement instrument properties. If the client−server connection is broken or remains idle over 5 min, the server application shuts down the system through Labtech. If the power shuts down, a system of safety interlocks in the physical system prevents the system from running indefinitely.



Feed Flow Valve: ON

Feed Valve Position 50 %

Tracer Feed Pump: ON

Injection Duration 2 sec

INJECT TRACER: TRACER READY

Run Time 128.900 sec

Feed Flow Rate 0.17608 lpm

Influent Conductivity 591.491 mS

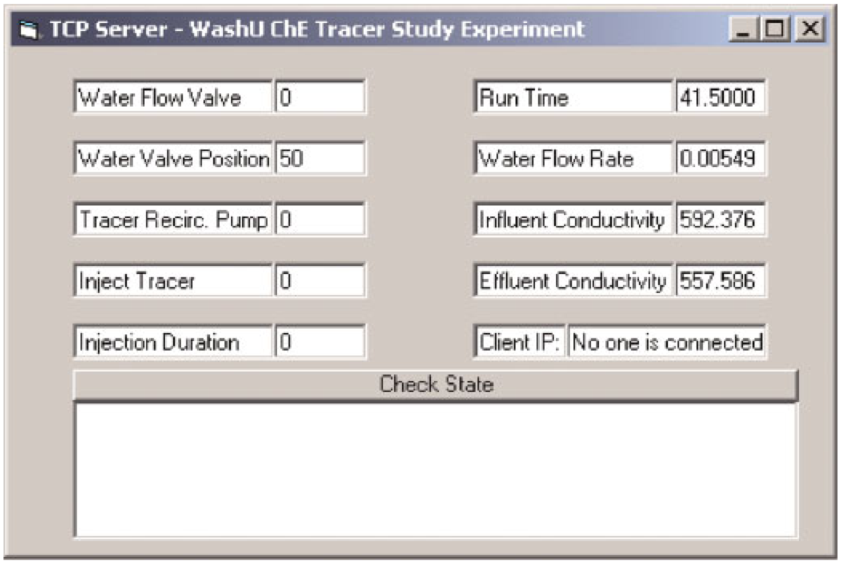
Effluent Conductivity 555.389 mS

Figure 6 Client application user interface. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

LAB EXECUTION PROCEDURE AND RESULTS

One team may operate the lab station either remotely via the Internet client−server-station connection or in the local way with traditional server-station connection. Through analysis of lab manuals and records, the lab session sequence have been identified and then revised based on the feedback from instructors and students.

A unit operation lab session using the proposed system is performed in three steps: a prelab for obtaining necessary background information on the project; the interactive real lab session executed by operating the equipment locally and remotely to collect data; and the final lab report including the analysis and evaluation of the data, as well as the rectification and diagnosis of mistakes. After a lab session is finished, the student is guided to appropriate online



Water Flow Valve 0

Water Valve Position 50

Tracer Recirc. Pump 0

Inject Tracer 0

Injection Duration 0

Run Time 41.5000

Water Flow Rate 0.00549

Influent Conductivity 592.376

Effluent Conductivity 557.586

Client IP: No one is connected

Figure 7 Server application user interface. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

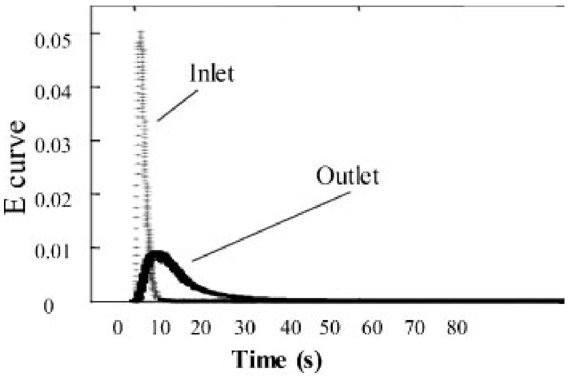


Figure 8 Typical tracer response curves shown in client side, collected at the inlet and outlet of the tubular reactor. Feed valve position: 90%, tracer injection duration: 1 s.

materials for more complete explanations and subject reviews.

The experiment is carried out so that students may analyze the mean residence time for the tubular reactor at a specific reactor feed flow rate by recording the changes in conductivity over time. The students open the reactor feed flow valve, start the tracer feed pump, control the flow rate by adjusting valve position, select the injection duration to accommodate the desired reactor feed flow rate, and then inject the tracer. The conductivity measurements will rise and then fall back to their steady state values, at which point the students may close the client application.

The students issue commands from the client computers to the server computer to monitor and control the experiments. Then the experimental results are collected, sent back by the server, and displayed in the client window. The client program also creates a log containing measurements of experimental variables. Once the client operation is closed, the client can open this log in Microsoft Excel to analyze the evolution of the collected tracer response. Typical tracer response curves at the inlet and outlet with respect to time are shown in Figure 8. Injecting a known quantity of the tracer and detecting the tracer responses allow the analysis of the reactor's characteristics. In an ideal plug flow reactor, the collected tracer curve would be a thin, spike-like peak. Comparing the response curve at the exit to the one at the entrance gives a measure of the reactor's nonideality, which is important when predicting reactant conversion from the given feed rate and composition.

Typical experimental results obtained from on-site local control and online remote control are listed in Table 2. The relative error between a local control run and its corresponding remote control run is determined for all six flow rates. The average absolute relative error of the latter to the former was found to be 2.91%. This insignificant relative error leads us to conclude that using on-site control or online control would have a similar effect on the residence time in the tubular reactor. Although there is a time delay between the client and server due to the web data transfer and the instrumentation synchronization, this delay penalty is negligible when instruments have to take a long time to complete the measurement procedure [7].

CONCLUSION

We have developed the remote control and measurement systems for conducting interactive online experiments via the Internet. The design structure and implementation approach are presented. The online laboratory allows geographically dispersed team members to collaborate in carrying out a given task. The established TCP/IP solution permits the server to control the experiments, record the measurements, and transmit data back to the client side. Because of their relative flexibility and compatibility, Visual Basic and Labtech technology are combined to construct the programs. The present system offers a well-structured development environment that could be widely adopted and expanded by the web community for different applications. The current lab scheme is dedicated to chemical unit operation, which includes the experiments performed in continuous flow reactors. It is used as a novel module in the chemical

Table 2 Comparison Between the Remote Control and Local Control

Valve opening position (%)	Actual flow rate (cm/s)	Local control (s)	Remote control (s)	Relative error (%)
45	2.93	22.96	23.77	−3.53
55	3.86	18.32	19.02	−3.82
65	5.11	14.57	14.97	−2.75
75	6.02	11.9	12.19	−2.44
85	6.44	11.74	11.38	3.07
95	6.67	10.87	10.67	1.84

reaction laboratory course for senior students at Washington University in St. Louis. The students are encouraged to conduct the lab sessions via both the remote access and the traditional local access. Such exercises offer valuable opportunity for students to gain insight and experience regarding the online process technology.

REFERENCES

[1] D. Shin, E. S. Yoon, S. J. Park, and E. S. Lee, Web-based interactive virtual laboratory system for unit operations and process systems engineering education, Comput Chem Eng 24 (2000), 1381–1385.
[2] W. Aung, P. Hicks, L. Scavarda, V. Roubicek, and C. H. Wei, Engineering education and research: A chronicle of worldwide innovations, iEER, Arlington, VA, USA, 2001.
[3] M. Hough and T. Marlin, Web-based interactive learning modules for process control, Comput Chem Eng 24 (2000), 1485–1490.
[4] I. T. Cameron, An interactive web-based decision support system for hazardous industry land-use planning, Comput Chem Eng 24 (2000), 1057–1062.
[5] D. Shin, E. S. Yoon, S. J. Park, and E. S. Lee, A Web-based, interactive virtual laboratory system for unit operations and process systems engineering education: issues, design and implementation, Comput Chem Eng 26 (2002), 319–330.
[6] H. M. S. Lababidi and C. G. J. Baker, Web-based expert system for food dryer selection, Comput Chem Eng 27 (2003), 997–1009.
[7] M. Bertocco, F. Ferraris, C. Offelli, and M. Parvis, A client–server architecture for distributed measurement systems, IEEE Trans Instrum Meas 47 (1998), 1143–1148.

BIOGRAPHIES

Jing Guo received his bachelor's degree in chemical engineering in 1997 and his master's degree in chemical engineering in 2000 from Beijing University of Chemical Technology. He received his PhD degree in chemical engineering from Washington University in St. Louis in 2005, where he worked on the experimentation of catalysis in a multiphase reactor, including a trickle bed reactor and packed bubble column. He also developed modeling program to simulate the multiphase reactions for applications ranging from bench scale to commercial scale.

David Kettler was awarded bachelor's degrees in both biomedical and chemical engineering at Washington University in St. Louis in 2001. During his study, he was also responsible for developing the Process Control Laboratory's homepage and the Simulink Virtual Laboratory as an interactive series of workshops.

Muthanna Al-Dahhan is a professor in the Department of Energy, Environmental and Chemical Engineering at Washington University in St. Louis and codirector of the Chemical Reaction Engineering Laboratory. He received his bachelor's degree in 1979 from University of Baghdad in Iraq, master's degree in 1988 from Oregon State University, and doctoral degree from Washington University in St. Louis in 1993. His research interests are in the fields of multiphase reaction engineering with development and implementation of advanced measurement techniques and modeling of transport-kinetic interaction for design and scale-up of multiphase reactors and processes for production of fuels and alternative fuels, chemical bioprocesses, biomass conversion, energy/bioenergy, and waste treatment. He is an author on more than 100 peer-reviewed papers.