Remote Laboratories and Team Skills in Mechatronics

James Trevelyan
School of Mechanical Engineering, The University of Western Australia

Abstract
A remote laboratories system built at the University of Western Australia provides a useful learning environment for mechatronics skills, particularly the combination of technical skills and project management. The remote laboratories framework was designed with this in mind and overcomes one of the main limitations of other approaches that seems to account for the difficulties in real implementations: the cost and complexity of technical support needed for development and maintenance. The students learning in the context of the project develop these skills. The design provides a technical environment that is simple enough for undergraduate students to understand and make effective contributions.

Keywords:
Remote laboratories, internet, mechatronics, professional engineering skills, team skills, technical skills.

1. INTRODUCTION
Many remote on-line lab experiments have been reported in the literature recently (for recent surveys see [5,6]). (Note that there are many reports of “virtual labs” where “virtual” means simulated. It is important to emphasise that this paper is entirely concerned with remote access to real equipment.)

From the first months of operation in 1994, our telerobot [15] was used from time to time, on request, as a device to help with courses on robotics. One of the earliest examples of a purpose designed lab experiment system on the web was described by Henry [9]. Salzmann et al [12] and Gillet et al [8] describe an internet-accessible DC servo device and how it can be used to help with student learning. Shen et al [14] and del Alamo [4] describe similar arrangements in which students can remotely test semiconductor devices in web-enabled experiments. A further similar arrangement has been developed in Norway (Fjeldy and Jeppson 2001). Röhrig and Jocheim [11] and Lemckert and Ferguson [10] and many others have worked specifically on distance learning applications of this technology. Qanser and National Instruments both offer commercial software and hardware for web-enabled control system lab experiments, and widespread engineering packages such as MATLAB and LabVIEW have well-developed software tools.
What has been learned from these pioneering efforts? Few of the early pioneering web sites are still in operation, or even show signs of recent developments. In terms of academic papers there was a surge of interest between 1994 and 2001, and the pace of development has slowed since then. We surveyed 61 remote lab projects in May 2003 and found 21 had disappeared or become non-functioning. Only 26 were on-line to users outside the developing department or research group. The telerobot site is the only one of these to have continuously operated throughout this period. The UTC web lab [9] has operated since 1995 but considerable efforts have been required for maintaining a modest system. Operating and maintenance costs have been significant: special purpose hardware-related software has been a major impediment to the maintenance of several projects. Maintaining software capabilities through significant operating system, computer hardware and user software upgrades has taken significant resources and is cited as an issue of concern by most authors.

High software investment and maintenance cost has not discouraged further development efforts, however. Several new remote labs are under construction at the time of writing, particularly in Western Australia, the USA, Germany and UK [1, 13 etc.]. The UWA system is being extended [17] and the MIT iLab project [4] is being re-implemented in a Microsoft .NET framework. Most current projects have devoted more resources to a remote lab framework: a set of tools that enable many different remote labs to be deployed without large additional investment in software and expertise. At the same time, the ready availability of commercial software has enabled many smaller institutions to set up remote labs on an individual basis.

The relatively small number of working systems and the slow rate at which this technology is progressing suggest reflect some significant problems with early implementations. The problems appear in the form of high installation, operation and maintenance costs. Some of the underlying causes are:

• Cost of maintaining the system to keep up with hardware, operating system, internet service provider and browser technology changes.
• Complexity of technological components and of integrating a working system.
• Lack of administrative support for large classes.
• Need to deal with unreliable internet connections.
• Need for collaborating users to be able to work together.
• Difficulties with integrating on-line lab experiences into conventional courses.

In 2002 we commissioned a comprehensive internet framework for remote access labs that aims to overcome these problems [17]. The experience of the telerobot project [2] contributed a reliable system design which became the basis for a new system built in the LabVIEW programming environment. Further details are available at http://www.mech.uwa.edu.au/jpt/tele/.

In this paper we will show how the remote laboratories system provides a useful learning environment for mechatronics students, while the students, at the same time, help to provide a reservoir of technical skills needed to develop and maintain the system.
2. SUPERVISORY CONTROL: A NECESSARY FRAMEWORK

Many remote laboratory systems have been implemented in Java. Dalton [2] implemented the second major revision of the telerobot software in Java. However, the complexity of such systems is considerable and demands highly developed programming skills and extensive experience with real time systems.

LabVIEW is another widely adopted alternative. Recent editions have included modules that greatly simplify the export of control panels to a remote Web browser. The technical difficulty of providing remote access to a LabVIEW application has been greatly reduced.

However, success with remote access depends on more than just being able to access the control panel with a Web browser.

There are two important sets of issues that need to be considered: technical issues relevant to supervisory control, and human factors relevant to the learning experience and the effective engagement between a person and a piece of equipment located far beyond reach. While they are important, human factors are complex and require more treatment in depth than is normally possible within the context of the course on mechatronics systems. However, supervisory control issues are core material in mechatronics.

Distributed control systems using Internet communication technologies are commonplace in industry and students of mechatronics need to be comfortable working with them. Supervisory control is only one of many approaches used in control system design. The supervisory control model consists of two parts: an autonomous local controller that receives commands from a remote user and implements them, and a remote operation system through which the user sends commands and sees the results. This model has proved to be effective for reliable Internet control equipment because the communication medium has intrinsic unreliabilities [2]. Remote users can lose connections and traffic congestion can add intermittent transmission delays.

![Figure 1: Instrumented remote access electric iron equipment at UWA](image)

**2.1 UWA Telelabs System Hardware**

The telelabs system is designed to handle any kind of experiment that can be operated remotely. Even experiments that require some manual intervention can still be operated remotely for part of the time. Therefore, it is assumed that the
equipment will be connected to a nearby computer in some way to exchange information and control settings. An example (electric iron) is shown in figure 1.

A high performance PC provides the server functions: it does not require special I/O devices so it can readily be replaced if it fails or needs upgrading. The server runs the main Labs on Line (LOL) Server application and the National Instruments DataSocket server.

Two types of clients are present in the system. Following Dalton [2] both the remote clients that users interact with as well as the hardware clients that supervisory control of the experiment hardware I/O connect to the server in the same fashion. Hardware clients have special status within the server, of course.

We make extensive use of FieldPoint (a proprietary field bus system) to provide a convenient way to provide an interface for the different hardware clients. Hardware clients can run on the same computer as the server, or can be distributed on several other computers if the server load needs to be reduced, or real-time I/O is needed. Local Fieldpoint modules provide interfacing (analog or discrete signals) for all devices requiring less than approximately 30 Hz sampling. FieldPoint 2000 allows real-time systems to be implemented at Fieldpoint modules where higher speeds or more precise timing is needed. High speed I/O requires special data acquisition cards in PC computers. The system is designed to be usable with any of these interfacing options.

Figure 2 Data flow between telelabs modules for iron experiment. Each process block represents a LabVIEW application: they normally run on different computers and communicate by internet TCP/IP socket connections.
Each piece of experiment hardware will have an adjacent PC workstation, sometimes equipped with web cameras for monitoring the rig remotely. These computers are needed when students operate the equipment in the laboratories during daytime hours. Apart from providing a computer workstation for the students to use, they allow for on-line documentation access, and they monitor student use of the system. For most experiments where Fieldpoint interfacing is possible, no special I/O interface hardware is needed so the workstation cost and specifications can be modest.

Currently the FieldPoint module and lab computers running the system are connected to a 100 MHz local network running on a switched hub. This prevents general building network traffic from interfering with data acquisition using Fieldpoint in the lab.

2.2 Telelabs Software

The software, in operation, consists of four distinct types of module, as shown in figure 2.

1) A hardware client controls the experiment hardware.
2) A remote client running on the remote computer: it communicates with the remote user, sends commands to the hardware client, receives results back and displays them to the remote operator. All users interact with the system with the remote client, whether working in the laboratory beside the equipment or using the system remotely.
3) A Labs-On-Line (LOL) server controls user access, decides which remote user currently has control (if several are collaborating on the one experiment) and provides administrative functions such as timetabled reservations, course and hardware configuration etc.
4) LabVIEW Data Socket (LDS) server for broadcasting status information to remote clients.

We are actually combining two concepts here: supervisory control and message-oriented middleware. The hardware client controls the rig in real time. The remote client simply serves as an operator interface that sends control commands to the hardware client (via the LOL Server).

The hardware client sends experiment results and current control settings to the data socket server to be broadcast to all remote clients.

The remote clients allow users to log into the server and control personal settings such as course enrolment details, contact details etc. The remote client also loads hardware-specific control panel libraries tailored for the task the student has to perform. Thus, a student might use one of a series of possible control panels depending on the particular task he or she has to perform. Some of the experiment control panels may be very simple and only allow very limited operations.

In both the remote and hardware clients, the part of the client that deals with authentication, displaying queue information, updating user details and operations common to all labs is contained within a common framework so that hardware-specific aspects are as simple as possible.

One of the remote clients is a special control panel for use by maintenance staff for fault diagnosis and testing purposes.
The remote client also includes a simulator so students can practice with the control panel while they are waiting in the queue.

The LOL server allows students to disconnect for some time without losing their place in the queue.

The system assumes that each student is enrolled in one or more courses that provide experiments in the system. Each course is associated with an individual (the course coordinator) who is responsible for creating the experiments, defining the tasks that students have to do, and monitoring student behaviour.

The concept of a ‘task’ arose some way into this development. Rather than considering a piece of experiment hardware as being associated to one experiment and a particular course, we can think of a task that a student is required to perform, that uses a particular piece of hardware and is associated with a course. Each course then consists of a number of separate tasks. A different control panel can be associated with each task so that the display of controls and data from the hardware can be tailored to the particular tasks the students are expected to complete. The same hardware can be used for different tasks, and even in different courses. For example, the same electric iron rig can be used in the context of courses on thermodynamics, mechatronics or control systems. In an introductory task, the remote client is simple and displays only the controls needed for the task. A more advanced task might have a completely different arrangement of controls and displays, and a client designed for use by maintenance personnel would be different again. Visitors can use the system by registering for a ‘tour for visitors’ course that allows short sessions on equipment that requires no prior training.

A remote access laboratory that provides reliable service must be able to withstand the effects of connection loss and transmission delay. The hardware client is able to operate the equipment for an extended period that can occur if a remote user loses his or her connection. While it cannot receive commands, it can continue to run an experiment, collecting the results for transmission to the user by Email at a later time. The hardware client needs to do much more, of course. After the user disconnects voluntarily, or the hardware client is notified by the server that the user seems to have decided to remain disconnected, the hardware client restores the equipment to a default condition, ready for the next user. At the same time, even when the equipment is not in use, the hardware client continually checks for hardware default conditions and maintains safety interlocks that protect both the equipment and people in the laboratory. The hardware client enforces restrictions that prevent the equipment from being damaged, either because commands have been misinterpreted all because users attempt to drive the equipment beyond safe limits.

We have implemented fault detection in a similar manner to that developed for the sheep shearing robot control system [16]. When a fault has been detected the hardware client notifies the experiment supervisor with an Email message. An important issue here is to ensure that the number of Email messages is strictly limited. If the hardware client generates too many fault indications supervising personnel will ignore them.

Another important issue for remote laboratories is collaborative access by more than one user at the same time. There are two scenarios: first, the student
may need assistance from an instructor and secondly some tasks are appropriate for
good work with more than one student working at the same time. We have
implemented collaborative access using two models.

All remote clients display the current control settings; these are broadcast by
the data socket server. Any user can modify the control settings. When this
happens the remote client transmits the new settings to the LOL server. As soon as
the server receives a request to change control settings it issues a status indication
to the other remote clients that makes them disable their controls for a fixed time --
typically about three seconds. During this time the inactive clients will receive the
new control settings and update their displays accordingly. If two clients happen to
send control requests at almost the same time the LOL server accepts the first and
ignores the second.

The second model simply extends the exclusive period of control. This is
appropriate in circumstances when inadvertent simultaneous control would result
in confusion. The telerobot is a good example of this.

The main task for students working on system development is the addition of
new experiment hardware and a preliminary set of remote client control panels. To
do this they need to perform the following tasks:

- Set up a stand-alone LabVIEW application to control the experiment
  hardware. This activity enables them to solve interfacing issues and
develop ways to display data from the experiment. They follow
templates developed for earlier experiments to structure their code in
appropriate ways.
- Connect the experiment hardware controller into a standard hardware
  client template and set up fault monitoring by defining allowable
  limits for important status indicators.
- Place the experiment display panels into a template for the remote
  client control panel library.

Staff then configure the server to allow the student to test the software in the
remote operating environment. Students can modify and restart their hardware
client when they want to without affecting the LOL server running all other tasks.

In summary, therefore, remote laboratories provide a challenging and
relatively complex application for supervisory control methods. They provide an
excellent learning environment in which students can acquire useful skills for
developing mechatronics systems. The question is whether the students have the
necessary competence, skills and background to be able to make effective use of
this learning environment.

3. EXAMPLE EXPERIMENTS
The following sections present some current examples of remote experiments to
illustrate this classification.

3.1 Instrumented Electric Iron
This experiment is offered currently at UWA. A domestic electric iron fitted
with temperature sensors and a controllable jet of compressed cooling air, can be
operated in several different ways:
• simple manual on-off control,
• pulse width modulated power control,
• feedback control.

The equipment can be used for several lab classes:
• Thermodynamics of a simple domestic appliance, heat transfer by convection and conduction.
• Modelling of a domestic appliance, from simple first order equation representation to finite element thermal modelling.
• Mechatronic discrete control and sensing.
• Control system theory applied to a simple non-linear system.

Each different kind of class has a different control panel adapted for the learning objectives.

Figure 3: Typical introductory task control panel for electric iron experiment

The lab experience requires a student to set certain operating parameters and interact with the equipment for a time of between 5 and 30 minutes depending on the tasks to be performed. Since the appearance of the iron does not significantly change there is little value in providing a real-time image of the iron for the student user.

The equipment is inexpensive. However, there are several aspects of its behaviour that are subtle: these can present a significant challenge to undergraduate students. For example, when using the internal thermostat, the temperature at which the thermostat switches off the heating element decreases significantly over the first 15 minutes after the iron is first switched on. This is not easy to explain, given that the thermostat is a temperature sensitive switch with well-defined switch on and switch off temperatures.
3.2 Torsional Vibrations
A servo motor excites low frequency rotational vibrations of two discs coupled by soft torsion springs. Students need to observe how different amplitudes and frequencies of excitation affect the motion of the discs. Data on disc motion arrives rapidly in real time (data is collected at 30 Hz) but students must observe the discs for several minutes at a time as transient effects last for up to two minutes.

An instrumental and chart display is sufficient to display the disc motion. While it is preferable that students can observe the discs directly using real-time video the data rate required means this is only feasible using broadband or local area networks.

Figure 4: Torsion Vibration remote access equipment at UWA: an example where high rate real-time data needs to be shown to the user.

4. EXPERIENCE WITH UNDERGRADUATE STUDENT CONTRIBUTIONS
Students have contributed to our remote laboratories project in two kinds of undergraduate units: second and third year project based units in mechatronics systems and mechatronics design and final year thesis projects. In the project units students work in mixed teams of second and third year students for 10 weeks (about 60 hours each in total). The third year students provide leadership and coordination as well as advanced technical skills while the second year students learn the fundamentals.

Several final year projects have resulted in substantial and successful contributions to the remote laboratories project. In some cases students have both designed equipment and implemented hardware and remote clients. However, they have found it difficult to complete all the aspects required for reliable 24 hour operation. Some students have focused almost entirely on software development and the educational issues and have produced very satisfactory working results that
we are currently using in teaching programs. Examples include the electric iron (shown above) and position control [3]. In the latter case the equipment resulted from two previous final year thesis projects.

In the project units we have more limited objectives. Typically students have managed to make significant enhancements to existing equipment rather than begin entirely original and new developments. Nevertheless they have been able to comprehend the system and connect existing equipment for remote operation. Some of the enhancements include developing additional equipment such as computer controlled brakes for the discs on a torsional vibration test rig.

Not all of the students in the project units work on the remote laboratories system: we allow students to choose from a wide range of projects working under several different supervisors. These range from small soccer playing robots to medical laboratory equipment.

In conclusion, we have not entirely solved the expertise problem. The equipment and software developed by a students tends to require some further refinement. However, this represents a substantial improvement from the previous situation when we relied on PhD students with substantial software skills. With time and improved documentation we expect the results from students to improve steadily.

5. EXPERIENCE WITH ON-LINE LAB CLASSES

To evaluate the utility of this system we offered third year mechanical and mechatronics engineering students an option to repeat part of the experiment they had performed in scheduled laboratory classes to improve their learning. Software for this particular laboratory was developed by a final year mechatronics engineering student who was also the teaching assistant supervising the laboratory classes [3].

These students were invited to use a remote laboratory task to explore aspects of controller tuning. The task required them to set given proportional, integral and derivative gains for a controller driving a large pointer in the laboratory, and measure performance parameters such as rise time and overshoot. The task was optional and set in the second last week of semester, so the relatively high number of students who attempted the task was very encouraging. The students were asked to answer an on-line questionnaire, and identified themselves by student number so that we could relate their responses to log file records.

67 students responded to the questionnaire, of which 62 of the students used the system. 57 students managed to operate the remote lab for more than 5 minutes. This attrition was due partly to inability to install or operate the software well enough to connect to the server.

Of the 57 students who used the system for significant lengths of time, the average total operating time was 21 minutes. Graph 1 shows the number of users who achieved given operating times. We see that most users achieved operating times between 10 and 30 minutes, though these were often in short sessions. The maximum time that a student was permitted to reserve was 15 minutes. Most sessions were less than 15 minutes in duration. For early users, a fault in the system limited sessions to 5 minutes and this affected about two thirds of the class.
This fault also caused some problems for users which limited the number of successful connections to the system.

![Operating times recorded for remote laboratory](image)

This result contrasts with the scheduled laboratory classes. 10 classes were scheduled for a total enrolment of about 112 students over a 4 week period. Attendance at the early classes was typically 6 – 7 students, and up to 15 students for the later classes. Although there was a booking system to limit attendance, in practice students forgot to attend earlier classes and so later classes were overcrowded.

Table 1 reports the responses by students to the on-line questionnaire. One of the questions they were asked was to recall how long they operated the equipment during the scheduled laboratory classes.

<table>
<thead>
<tr>
<th>Operating time reported by students</th>
<th>Number of students</th>
</tr>
</thead>
<tbody>
<tr>
<td>Watched, did not operate</td>
<td>28</td>
</tr>
<tr>
<td>&lt;10 minutes</td>
<td>8</td>
</tr>
<tr>
<td>10 – 20 minutes</td>
<td>14</td>
</tr>
<tr>
<td>20 – 30 minutes</td>
<td>5</td>
</tr>
<tr>
<td>30 – 60 minutes</td>
<td>6</td>
</tr>
<tr>
<td>&gt; 60 minutes</td>
<td>5</td>
</tr>
</tbody>
</table>

These results demonstrate that the remote lab has significantly extended the laboratory experience for most of the students in the class. Only 16 students (about 25%) operated the equipment for 20 minutes or more in the scheduled laboratory class. However, all except 14 students managed at least 10 minutes using the remote laboratory, and most managed more than this. Nearly all the students managed to complete the assigned task using the remote laboratory.

Other questions explored student preferences. We found that a significant number of students had difficulty installing the software in our computer laboratories, and
at home. Installation CD’s were made available but few students used them. A simple download and install procedure is needed. However, with more labs on the system, students would learn how to do this better. Most of the students said they preferred to use the real equipment if was available. When asked why they did not operate the equipment in the laboratory, most said this was because others were operating, or there was not enough time for their turn in the laboratory class. Some said they did not know what to do (10), or were afraid they would make a mistake in front of other students (6).

In another class, an on-line laboratory was made available for second year mechatronics engineering students to develop a simulation model of an electric iron. This exercise was conducted in the last week of semester. The students did not have to use the on-line laboratory because data from the lab was provided independently to the class, but extra marks were awarded for a simulation model that would work in cases other than the one supplied. Out of 50 students in the class, 21 students elected to use the on-line laboratory for an average time of 53 minutes (cumulative). Average session time was 16 minutes: the maximum possible reserved time was 15 minutes, but many students were able to achieve longer session times because they used the equipment at periods of low demand. Peak usage time corresponded to attendance at the University – 12 noon till 6 pm.

Unlike the position control experiment, most of the students had never operated the equipment before. Most had seen the equipment in the laboratory during other class activities.

Around 75% of all access to the system was from on-campus computer laboratories and 60% of these sessions were from our own computer laboratories where the software had been pre-installed. Interestingly, even though students could use the equipment from adjacent computers for much of the day when the laboratory was open, almost no students chose to do so. Had the system been used for larger classes with tighter deadlines, we could more students would have used the system from off-campus locations. Some students complained about having to download the initial installation files (12 Mbytes). However, once the initial installation has been accomplished, each new laboratory client only requires a 1.5 Mbyte download. The electric iron experiment involved quite long operating times because it takes time to collect the required data. Students could happily work on other assignments while watching the temperature chart.

Even though they were free to do so, none of the students chose to operate with other students at the same time. We did not draw the attention of students to this facility. On at least one occasion a server fault enabled two students to operate the equipment at the same time accidentally. The students reported that “someone was hacking into the system”.

**Conclusions from 2002 Evaluation**

Our first trial with remote access laboratories showed that students obtain more operating experience in a week than over 5 weeks of normal laboratory classes. Further, in the laboratory classes 53% of the students used the equipment for less than 10 minutes, and about 40% of the students never operated the equipment at all. Using the remote laboratory system, 80% of the students managed to complete
the required tasks, operating the equipment for themselves. This figure would have been higher but for some outstanding software issues that were resolved in 2003. Although it has taken three years to develop this system we are very happy with the results and it will become a standard part of our learning environment for students.

2003 Classes
In the current year we have operated an experimental series of on-line lab exercises for a class of second year mechatronics students that form the heart of tutorials over a period of 8 weeks. They use the electric iron equipment to study the control of a simple consumer appliance, and then how to model its thermodynamic behaviour. Finally they use the model to develop alternative improved control methods using pulse width modulation to control heating and cooling of the iron. The on-line lab brings real world data (and inconsistencies) into the tutorial problems and the students seem to respond well. Evaluations will be performed before the class finishes.

6. ACKNOWLEDGEMENTS
This work was supported by the Faculty of Engineering, Computing and Mathematics and the University Initiatives Fund. The author thanks his many colleagues who contributed parts of the technical work reported in this paper. Further details are available at [http://www.mech.uwa.edu.au/jpt/tele/](http://www.mech.uwa.edu.au/jpt/tele/).

7. REFERENCES