Web Based Telerobotics and Hardware-in-the-Loop Simulations as Enhancements to Mechatronics Teaching

E D Lindsay, M C Good & S K Halgamuge
Mechatronics Research Group
Department of Mechanical & Manufacturing Engineering
The University of Melbourne, Victoria, 3010, Australia
[edl, mcg, sam]@mame.mu.oz.au

Abstract
The advent of the Internet has revolutionised modern society, and provided educators – particularly in fields such as mechatronics - with previously unimagined degrees of flexibility in the delivery of their material. Capitalising upon this flexibility whilst retaining credibility and authenticity can sometimes be a challenge, with the perception of reality having to give way to the "virtual" equivalent. At the University of Melbourne's Department of Mechanical and Manufacturing Engineering, this challenge has been addressed with the use of a hardware-in-the-loop approach and, more recently, capitalising upon the Internet as a telecommunication medium to integrate web-based robotics into the curriculum. This has served to improve levels of student achievement, retention and enjoyment within the mechatronics courses to significantly above departmental and university wide norms.

Keywords
Education, Hardware-in-the-Loop, Java, Mechatronics, Telerobotics

1 Introduction

The field of telerobotics has seen increasing scope for its application in recent years. Whilst initial efforts were directed towards using telerobotics as a solution to previously intractable problems, such as operation in hazardous environments [1], more recently telerobotics has been considered as a method for improving existing robotic solutions.

The rapid explosion of growth in the Internet has provided the field of telerobotics with a convenient control medium. The Internet offers ubiquity and low establishment costs which cannot be matched by traditional dedicated systems, such as tether cables. There are a number of facilities to remotely control hardware through the Internet (eg [2] [3] [4]), but the majority of these sites are primarily research in nature.

Internet-based teaching resources are increasingly being integrated into curricula at the University of Melbourne. Multimedia materials and the World Wide Web are forming the backbone of many courses, and the Department's incorporation of actual hardware into this software-based teaching environment has made further significant improvements in the quality of its mechatronics teaching.

2 Mechatronics Teaching On-line

Preparing assignments for students to complete is often a challenge. The task must have sufficient content to keep the students occupied, and to test their learning, but must also be sufficiently small that the students can adequately complete the exercise. Students respond better to and achieve more highly on assessment tasks that they feel are authentic, "rather than contrived, artificial or decontextualized" [5], and this must be reflected in the way assessment tasks are constructed.

Hardware is more tangible than software, and the inclusion of hardware elements into a simulation – the hardware-in-the-loop approach – reduces the artificiality of the exercise, promoting a more interesting feel to the students. The broader, real-world applications of their work become more readily apparent, increasing the perceived relevance of the work.

The Department has been including the hardware-in-the-loop approach in its undergraduate teaching program since 1997. Projects since then have included an active truck suspension, and web-based telerobotics. As is often the case with more authentic tasks [6], the students found that the aims of these assignments were more ambiguous than they would prefer. This was offset, however, by a significant improvement in the level of intellectual stimulation provided by the assignment work, and an improvement in the extent to which the subjects were regarded as "well taught".

3 Active Truck Suspension

This project [7] is concerned with the design and implementation of a microprocessor-controlled "semi-active" damper to be used in the suspension of the drive axle of a heavy truck. Active and semi-active suspensions are being developed in the transport industry as a means of improving the comfort, safety and commercial viability of trucks, and of reducing road damage due to dynamic tyre forces. The project has four main components:

1. mechanical design of a damper capable of providing a continuously variable dissipative suspension force.
2. software design of a microcontroller program to acquire and process sensor data and to control the semi-active damper.
3. implementation of the program on an M68HC11 microcontroller in a "hardware-in-the-loop" (HiL) system simulation.
4. experimental investigation of the optimum control parameter settings for a given roadway roughness profile and vehicle speed.

The vehicle-suspension-roadway system is simulated with a parameterised Simulink™ model. Design parameters for the hydraulic damper, and for the PID damper-force controller, are supplied by the student investigator. The simulation interacts in real time with the microcontroller, generating analog "sensor" signals and accepting damper force commands. To assist students with the development of their designs, several Simulink models are provided [8]:

1. a simulation of the hardware-in-the-loop system itself, allowing students to test their data filtering, PID force control, and semi-active force command algorithms before implementation in the HiL system.
2. a tool to allow exploration of the effect of different "trim" positions of the nonlinear suspension system on the performance of the PID controller, eg by conducting Ziegler-Nichols "ultimate sensitivity" tuning experiments.
3. a simulation of a system with a passive damper, to allow evaluation of the potential benefits of the semi-active approach.

Student feedback indicated that the assignment is challenging but rewarding. They appreciated the quality of the online resources available to investigate the problem at a sophisticated level.

4 Implementing Telerobotics – The Java Advantage

Previously, the field of web-based telerobotics has been reliant upon the Common Gateway Interface (CGI) for transmitting information via the Internet [9]. The Department, however, has broken from this convention and developed the system using a client / server architecture in the Java programming language.

The system offers several advantages over previous telerobotic interfaces [10]. The object-oriented Java programming language is especially suitable for this application. Object Oriented code is more modular, and easier both to maintain and upgrade [11 Pressman]. This also leads to more a more flexible system, with the functionality of the system able to be adapted by the simple expedient of introducing alternative modules into the code.

The client side is implemented using Java applets. Modern Internet browsers provide significant support for the Java language, allowing the applet to be downloaded and run on the client's machine, reducing the computational load on the server.

The server side is implemented through the use of Java servlets, offering improved fault tolerance. Java servlets are unable to crash the server, allowing the system greater capacity to recover from errors. The platform independence of the Java system ensures that the server is able to reliably communicate with clients, regardless of the client's choice of operating system.

4.1 The Client Side

The client's applet interface window (Figure 1) provides the user with an image of the robot (1), numerical feedback as to its configuration (2), and also provides the user with a feedback window to which text responses can be sent (5). The user enters commands in the input text field (4), and is given a toggle checkbox to overlay a grid on the image (3).

When a command is entered, the first step is to ensure that the command is syntactically correct – that the sequence of characters entered does in fact represent an executable command. Syntax checking is handled by a dedicated Java class. Commands are parsed as a series of substrings separated by whitespace, the first

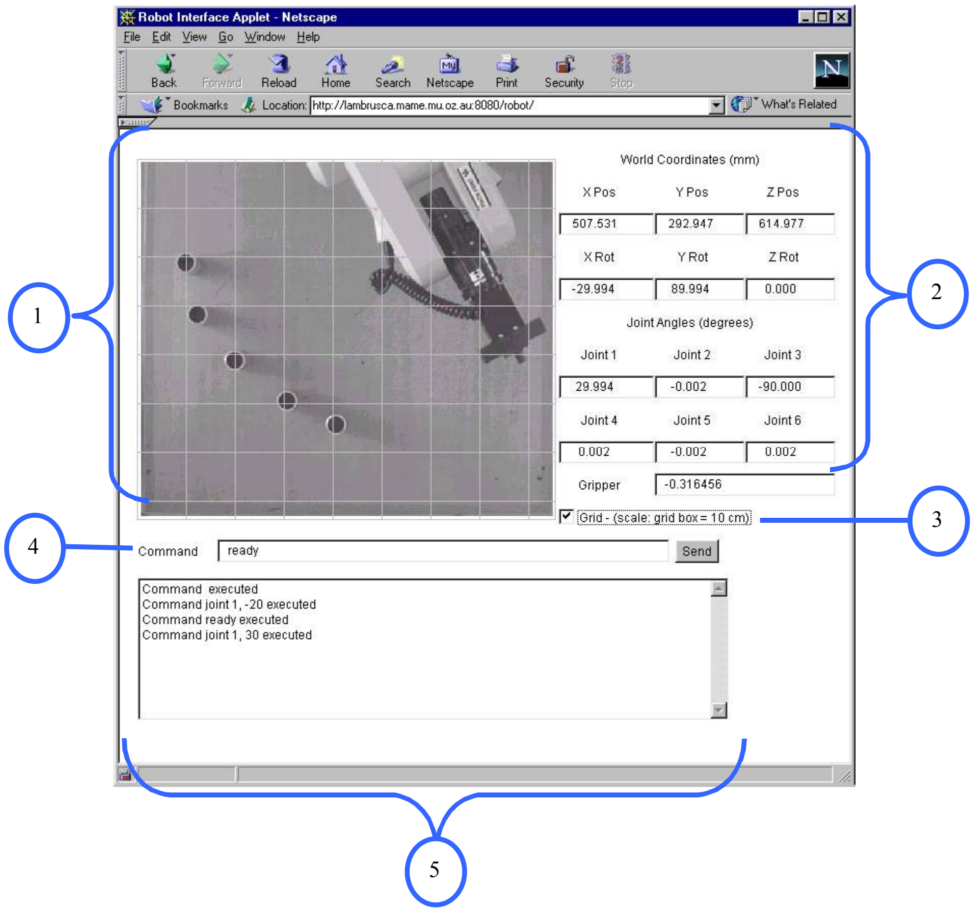

Figure 1 : The Telerobot Interface [10]

representing the command itself, with subsequent substrings representing the arguments to this command. The syntax checking class contains a list of valid commands – the dictionary of the robot's language - along with the number and type(s) of arguments accepted by each command. The input command is compared against the dictionary, and the command is either accepted, or rejected.

Once a command has been validated, it is sent via the Internet to the server, or in the case of interface-specific commands (such as the user help commands), implemented directly. Commands that are rejected lead to an error message being sent to the interface display.

Explicitly defining this dictionary offers an additional level of robustness within the system. Instructors can choose to only offer a subset of the entire functionality of the robot by only including the functions they wish remote users to have available. Similarly, instructors can offer a superset of the entire command language, offering additional functions – to this stage mostly based around manipulating the interface – by including them within the dictionary.

4.2 The Server Side

Whilst the syntax checker on the client side is responsible for verifying that the input string is a command, it is the task of the server side to interpret this command in context, and determine whether it should actually be implemented. Self-destructive commands must not be executed.

The server is also responsible for handling communication with the robotic hardware via serial connections. Again, this functionality is handled through dedicated Java classes, or in the case of the digital camera, by additional image capture software on the server machine.

5 The Next Step

The University of Melbourne places considerable emphasis on “transforming the curriculum” by taking advantage of new technological and pedagogical methods [12]. As a consequence, the mechatronics teaching program is constantly evolving. The following features will soon be added to the Telerobot system:

5.1 Standard Intermediate Language

One of the initial barriers to effective operation of any robotic machinery is the need to become fluent in the proprietary operating language of the equipment. Using the same standard dictionary of commands, regardless of the actual hardware, allows users to trivially adapt to a change in the physical hardware. Implementing this standard dictionary is a matter of changing the client side syntax checking class, and adding an “interpreter” class to the server side.

5.2 Improved Flow of Control

The ability to create more complex command routines allows for a greater level of complexity to be explored by the users of the system. The inclusion of enhanced batch processing of multiple commands, along with the possibility of if-then-else style flow of control, will allow a much greater degree of flexibility to be achieved.

5.3 Robot Simulator

The vast numbers of potential users for web-based robotics can lead to significant competition for access to the hardware. To assist in mitigating this demand, a Java Class to simulate the robot will be incorporated into the client side applet. Using this simulation, novice users are able to first familiarise themselves with the command interface that authentically represents the hardware, and once they have reached some level of proficiency to then make requests of the hardware itself. It is anticipated that this bias towards proficient users will lead to an increase in productivity from the hardware.

6 Conclusions & Discussion

The University of Melbourne’s Department of Mechanical & Manufacturing Engineering has successfully adapted to the use of online delivery for teaching materials. It has further enhanced this innovation by the introduction of actual hardware elements into these materials – using the hardware-in-the-loop approach to raise teaching standards further.

Whilst students find that the assessable tasks using this technology are more ambiguous, this setback is more than compensated for in terms of student motivation and enjoyment of the subject, and an overall increase in the standard of teaching within the subject. The feedback from the mechatronics projects exceeds Faculty-wide norms by a significant margin.

In the process of developing these teaching materials, the Department has created a number of useful mechatronics tools. The Department has successfully implemented effective web-based telerobotic control of a CRS F3 robot using the software developed in the Java programming language. This software offers performance benefits over existing CGI-based comparable systems, both in reduced delays for users, and easier and more flexible maintenance for the developers.

Acknowledgments

The authors acknowledge the University of Melbourne TaLMET Committee, who have partially supported this work through TaLMET grant number 1999R2AMUP098. The authors wish to thank the Advanced Engineering Centre for Manufacturing for partial funding of the hardware. The authors acknowledge Kid Kalanon, Kaz Fukuda and Bill Ross for their involvement in the project.

References

1. J Bares & D Wettergreen, *Dante II, technical description, results, and lessons learned*, International Journal of Robotics Research v 18 n 7 1999
2. University of Western Australia Telerobot (http://telerobot.mech.uwa.edu.au)
3. Cargnegie Mellon University's Xavier Mobile Robot: (http://www.cs.cmu.edu/afs/cs.cmu.edu/Web/People/Xavier/)
4. PumaPaint (http://yugo.mme.wilkes.edu/~villanov/)
5. J Herrington & A Herrington, *Authentic Assessment and Multimedia: how university students respond to a model of authentic assessment*, Higher Education Research & Development, Vol. 17, No. 3, 1998
6. S Butler, *Catalysing Student Autonomy through action research in a problem centred learning environment*, Research in Science Education, Vol 29, No 1, 1999
7. Project Webpage, http://www.mame.mu.oz.au/~mcg/mdl3b/
8. http://www.mame.mu.oz.au/~mcg/mdl3b/Simulink_models.html
9. K Taylor & B Dalton, *Issues in Internet Telerobotics*, [Conference Paper] FSR 97 International Conference on Field and Service Robotics, 1997
10. K Fukuda & K Kalanon, Poster Presentation at the 1st IEEE Workshop of Cluster Computing, December 2nd-3rd, 1999, Melbourne, Australia
11. R Pressman, *Software Engineering: A Practitioner's Approach*, 4th Ed. McGraw Hill 1997
12. http://talmet.unimelb.edu.au