Proc. 4th National Conference on Applied Mechanisms and Robotics, Cincinnati, OH, Dec. 10-13, 1995, pp. AMR-50-01/04.

Hypercollaborative Human-Machine Interaction through the World Wide Web

Harry H. Cheng, Assistant Professor
Integration Engineering Laboratory
Department of Mechanical and Aeronautical Engineering
University of California
Davis, CA 95616

Abstract

The hypercollaborative human-machine interaction is characterized by its hypercollaborative nature. Unlike conventional teleoperation, in a hypercollaborative human-machine interaction environment, mechatronic systems at a distant location can be controlled remotely by *different people* thousands of miles away at *different locations* through the multimedia/hypermedia user interface. A prototype hypercollaborative human-machine interaction system can be collaboratively controlled remotely from any part of the world through web browsers on the World Wide Web by linking to the Uniform Resource Locator address `http://iel.ucdavis.edu` for a hypercollaborative laboratory. The system is integrated under the C^H programming paradigm.

1 Introduction

The hypercollaborative human-machine interaction (HHMI) allows people at distant locations to use shared networks for collaborative real-time remote and automated control of mechatronic systems and instrumentations. It has numerous applications. For example, (1) HHMI is important to the infrastructure of manufacturing enterprise and agile manufacturing. Machineries such as CNC manufacturing machines and industrial robot manipulators can be controlled and their performance can be monitored remotely through HHMI. (2) As more and more devices are put in space for military, commercial, and exploration use, it becomes more and more desirable for humans to perform sophisticated manipulation of mechatronic systems for maintenance, for position of sensors, for data acquisition, and for carrying out scientific experiments in space entirely from the ground. It is also desirable that devices in space can be collaboratively manipulated and controlled remotely by people thousands of miles away at different ground locations as if they were in the remote unmanned space station, which will result in great savings in dollars and risk to life. (3) There are a variety of one-of-the-kind research facilities located in various national laboratories. The HHMI can optimize the use of these scarce facilities and enhance human collaboration. It allows routine and frequent participation of researchers in academia, scientists in NASA and national laboratories, and key personnel in industry in planning and conducting experiments in national labs, which otherwise require long-in-advance flying and housing arrangements, and site safety and operational training, etc. (4) The HHMI can find its applications in waste site clean-up and environment restoration. (5) The HHMI can also find its applications in undersea exploration and maintenance of offshore petroleum industrial facilities from onshore sites. (6) The HHMI can be used for intelligent highway maintenance and construction, especially for repairing of highway damaged by natural disasters such as landsliding. (7) The HHMI can also be used for underground mining remotely from the surface, which will take the worker away from the dangers of hazard minerals and deep mines. (8) The HHMI is critical for remote experimentation in distance learning. (9) An increasing number of companies are opening branch offices in cyberspace on the World Wide Web. The HHMI will allow customers interactively selecting merchandise in cyberspace as if they were shopping in a supermarket. (10) The HHMI can even be used in our daily life. The consumer electronical devices and household appliances can be controlled from offices at work and networked computers anywhere in the world as one is traveling.

2 Human-Machine Interface through the World Wide Web

To collaboratively operate instrumentations remotely from different parts of the world, a uniform user interface is desired. We choose the World Wide Web for user interface between machineries and collaborators because of its world-wide popularity. A sample web page for human-machine interface is shown in Figure 1. Using this hypermedia user interface, instructions for operation of machineries can be easily posted. For example, it is shown in Figure 1 that in order to use graphical user interface and monitor the performance remotely, the remote host name `iel.engr.ucdavis.edu` in our Integration Engineering Laboratory shall be added to the access control list by command `xhost iel.engr.ucdavis.edu` in the X-Window system at the local host. Essentially, the hyperlink at the human-machine user interface web page will be pointed to the `telnet` resource for hypercollaborative human-machine interaction. Each collaborator is assigned user account and password. Alternatively, accessing to machineries can be controlled by restricted access of web pages in a web server, which, depending on implementation, can be more secure

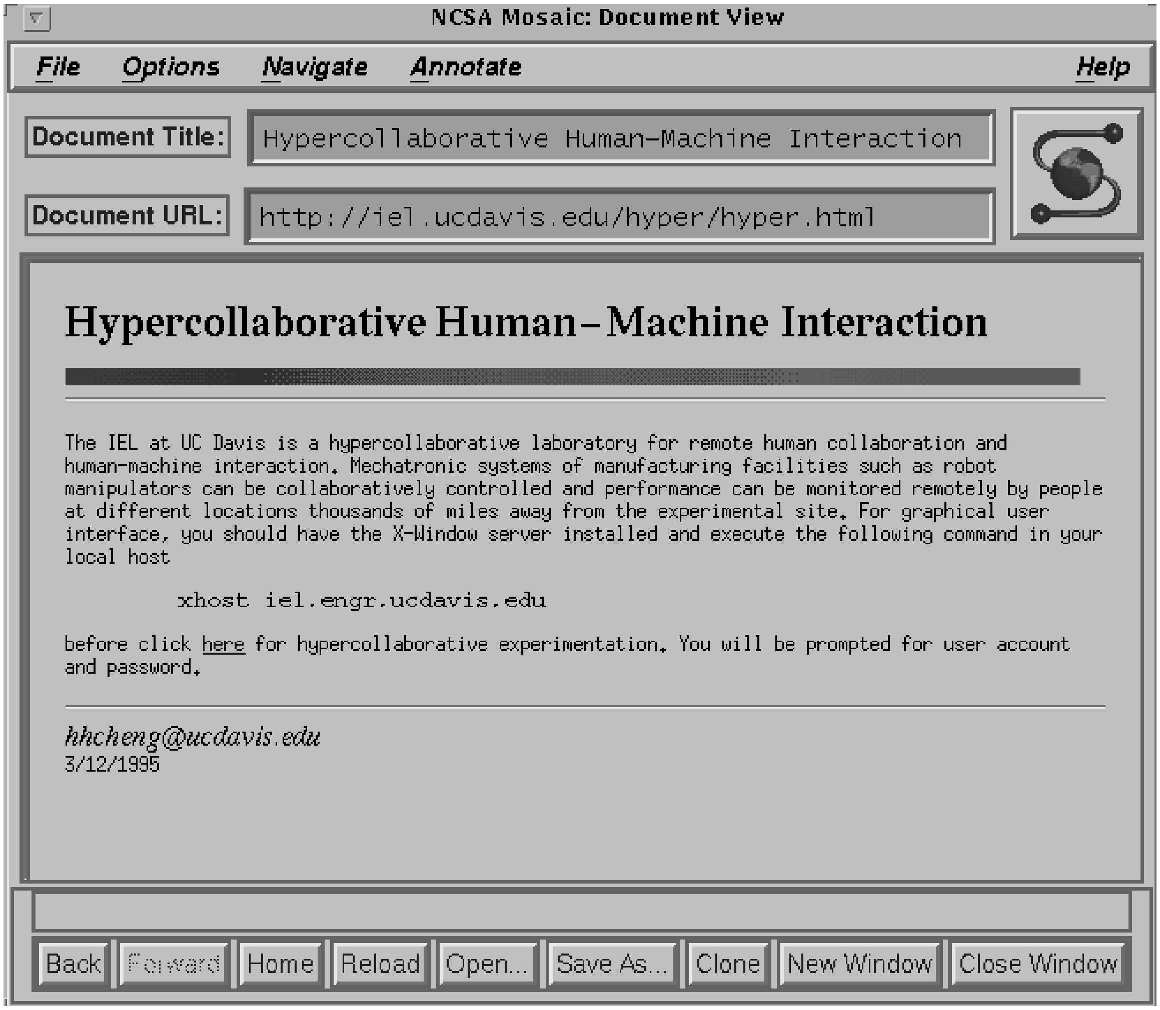

FIGURE 1: A web page for human-machine interface on the World Wide Web.

than using a password through telnet. For example, a web page can be protected by password and network domain. Restricted access of web pages by a network domain, which denies access from computers outside an organization, is useful for making the real-time manufacturing information available throughout the entire organization.

3 Real-Time Unix Operating System and Login Shell

Unlike conventional remote teleoperation where instrumentations are controlled by an operator from a single site on the ground [1, 2], hypercollaborative human-machine interaction involves many operators at distant locations. Distributed real-time computing is essential in hypercollaboration. For openness of system [3], a real-time Unix operating system is used in our prototype hypercollaborative human-machine interaction system. A Unix operating system compliance with POSIX 1003.1, 1003.4 and 1003.4a standards will ensure real-time performance and the portability of the developed code to a variety of popular platforms. Unix standards provide a reliable and well-known programming interface [4]. TCP/IP and NFS enable many networking possibilities based on ethernet. X-Window and Motif support allows direct implementation of graphical user interfaces and monitoring.

One of Unix's most important components is its user interface, or *shell*. Conventional Unix shells such as C shell [5], Bourne shell [6], and Korn shell [7] are inadequate for hypercollaborative human-machine interaction. Therefore, C^H shell has been developed for user interface of real-time Unix. C^H shell is similar to C shell in handling of variables [8, 9]. However, unlike the so-called C shell, C^H is designed to be a superset of C [10, 11]. It is more convenient to program in C^H shell than in C shell or C language [12, 13]. Besides, C^H shell's low-level features, retained from C, make the hardware readily accessible to operators. C^H shell can be used as a login shell of the operating system. A system administrator at the experimental site can restrict users by modifying the system start-up files `chshrc` and `chlogin` as well as each user's local start-up file `.chshrc` and `.chlogin`. Even keywords and system built-in variables can be removed by the built-in function `remkey(char *keyword)` and `remvar(char`

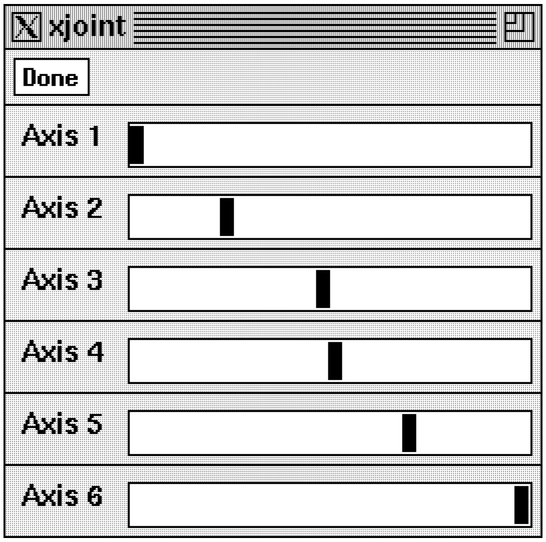

FIGURE 2: The graphical user interface for control a Puma 560 in joint space.

`*variable`), respectively, which is more flexible than other conventional shells. Like other shells, a restricted version of C^H shell will be implemented in the future for more restricted access of machineries.

4 Graphical User Interface and Remote Monitoring

Mechatronic systems such as robot manipulators can be controlled by C^H programs or commands under C^H shell. For some tasks, it can be more conveniently done through graphical user interface [2]. For example, each joint of a Puma 560 with six degrees of freedom can be moved by the graphical user interface shown in Figure 2. By dragging the cursor on an axis shown in the computer screen, a joint of the Puma 560 will move accordingly. We have developed a set of graphical user interface utilities for hypercollaborative human-machine interaction. In hypercollaborative human-machine interaction, multiple collaborators will operate the same mechatronic devices remotely. The priority and accessibility of mechatronic devices are decided by mechatronic device drivers. The permission of a mechatronic device can be specified when it is opened by standard I/O function `open()` and later controlled by I/O function `ioctl()` [9]. For example, multiple operators cannot drive a robot simultaneously.

On the other hand, some mechatronic devices can be accessed simultaneously by multiple processes. For example, the current position and velocity registers in a servo control board can be read by multiple processes. We have developed a set of utilities for performance monitoring. Figure 3 shows the monitoring of the forces and torques at the end-effector of a Puma 560.

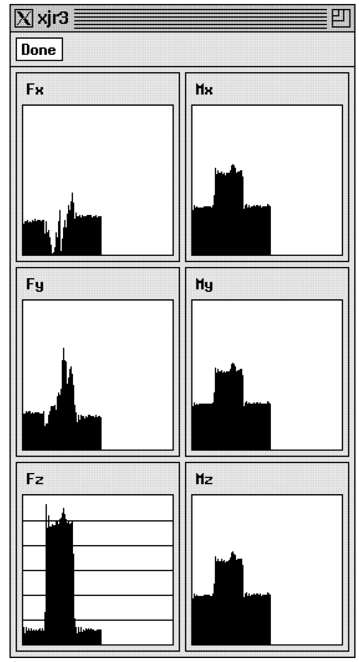

FIGURE 3: Monitoring forces and torques at the end-effector of a Puma 560.

5 Experimentation

A prototype hypercollaborative human-machine interaction system with multiple robots has been developed at the Integration Engineering Laboratory of UC Davis. The hardware configuration of the experimental setup is shown in Figure 4. We have retrofitted the Puma 560 controller in Figure 4. The retrofitted robot controller consists of servo controller, I/O and A/D interface boards from Delta Tau Data Systems, machine vision system from Datacube and Panasonic, force/torque sensing system from JR3. The system is coordinated by a C^H shell under a real-time Unix operating system LynxOS from Lynx Real-Time Systems running in a VMEbus based computer MVME167 from Motorola. There are several computer workstations, X-terminals, and PCs in the lab to drive robot manipulators hypercollaboratively by multiple operators. We also demonstrated the hypercollaborative concept by remotely control the system with multiple collaborators at different parts of the world. Our hypercollaborative robotic system is just one of ever increasing network devices on the information superhighway.

6 Conclusions

A hypercollaborative human-machine interaction through the World Wide Web has been demonstrated by remotely control mechatronic systems at the Integration Engineering Laboratory of UC Davis with multiple collaborators at different parts of the world. We are fine-tuning this prototype hypercollaborative experimental system. To achieve location transparency for hypercollaboration, we are adding multimedia communication among collaborators.

FIGURE 4: A prototype hypercollaborative human-machine interaction system.

7 Acknowledgements

This work was supported in part by the National Science Foundation as a Research Initiation Award under Grant DMII-9309207, by the University of California, Davis through faculty research grant program and by Datacube, Delta Tau Data Systems, Intel, JR3, Lynx Real-Time Systems, Motorola, Panasonic, and Unmanned Solutions. The author would like to thank K. Van Katwyk and J. Pannu for their contributions to the project.

8 References

1. Sheridan, T. B., Space Teleoperation through Time Delay: Review and Prognosis, *IEEE Trans. Robotics and Automation*, Vol. 9, No. 5, October 1993, pp. 592-606.
2. Gertz, M. W., Stewart, D. B., and Khosla, P. K., A Human-Machine Interface for Distributed Virtual Laboratories, *IEEE Robotics and Automation Magazine*, Vol. 1, No. 4, December, 1994, pp. 5-13.
3. Wright, P. K., and Greenfeld, I., Open Architecture Manufacturing: the Impact of Open-System Computers on Self-Sustaining Machinery and the Machine Tool Industry, *Proc. Manufacturing International' 90*, ASME vol. 2, March 1990, pp. 41-47.
4. Ritchie, D. M. and Thompson, K. L., The Unix Time-Sharing System, *Commun. ACM*, vol. 17, No. 7, July 1974, pp. 365-375.
5. Joy, W., *An Introduction to the C Shell*, Department of Electrical Engineering and Computer Science, University of California at Berkeley, 1980.
6. Bourne, S. R., The Unix Shell, *The Bell System Technical Journal*, vol. 57, No. 6, July-August 1978, pp. 1971-1989.
7. Korn, D. and Bolsky, M., *The Korn Shell Command and Programming Language*, Prentice-Hall, 1988.
8. Cheng, H. H., C^H Shell for Integration of Mechatronic Systems, *Proc. of the 1995 NSF Design and Manufacturing Grantees Conference*, San Diego, CA, January 4-6, 1995, pp. 95-96.
9. Cheng, H. H., Toward Task-Level Robot Programming, *IEEE International Conference on Robotics and Automation,* Minneapolis, MN, April 22-28, 1996 (under review).
10. ISO/IEC, *Programming Languages - C*, 9899:1990E, ISO, Geneva, Switzerland.
11. Kernighan, B. W. and Ritchie, D. M., *The C Programming Language*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1st edition, 1978; 2nd edition, 1988.
12. Cheng, H. H., The C^H Programming Language for Task-Level Robot Programming, *Proc. of the 1994 NSF Design and Manufacturing Grantees Conference*, Cambridge, MA, January 5-7, 1994, pp. 143-144.
13. Cheng, H. H., Extending C and FORTRAN for Design Automation, *ASME Trans., Journal of Mechanical Design*, Vol. 117, No. 3, 1995, pp. 390-395.