Int. J. Engng Ed. Vol. 12, No. 2, p. 115–122 1996
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Using the Internet to Share a Robotics Laboratory*

G. McKEE
The University of Reading, Department of Computer Science, Whiteknights, Reading RG6 2AY, UK
R. BARSON
University of Nottingham, Department of Manufacturing Engineering and Operations Management, University Park, Nottingham NG7 2RD, UK

Netrolab is a funded project which aims to demonstrate the use of the Internet to allow higher education institutions to share robotics resources which are otherwise expensive to set up and maintain. The robotics resources are housed in a laboratory, which we refer to as Roboscape, located at the University of Reading, and will be accessible to support teaching and project work in robotics and artificial intelligence subjects. We outline the requirements we are aiming to satisfy in order to provide this support through Netrolab, and we present the hardware and software infrastructure that we are building to allow the flexible creation of a wide range of educational modules. We present a case study demonstrating both flexible access to the robotics resources in the laboratory and the use of computing workstation laboratories to realize a virtual robotics laboratory. Netrolab represents a new opportunity arising from the combined power of high bandwidth networks and multimedia workstation technology to bring otherwise inaccessible resources to a wider audience than hitherto possible.

INTRODUCTION

THEORY and practice need to be mutually harnessed to achieve a rich educational experience for undergraduate students. Practice is achieved through individual and group laboratory experiments and project work. However, laboratory facilities place a significant burden on educational budgets, demanding space, equipment and technician support. In the face of these demands, emphasis is often placed on low-cost, replicable experimental equipment and resources which support the general theory of the subject, but leave the more specialized areas poorly, if at all, resourced. A consequence of not having laboratory resources is that theoretical coverage may suffer.

Few education institutions can provide laboratory support for robotics teaching because of the expense of setting up and maintaining a robotics facility [1]. Providing a minimal, one-station environment comprising a manipulator, a mobile robot, vision sensing and computer support can cost of the order of £70,000. Thus the teaching of robotics is often unsupported by practical laboratory experiments. This makes it infeasible for higher education institutions to provide effective training in robotics technology, hence stifling the conveyance of robotics knowledge and skills to the students in the first instance, and through the students into business and industry in the second instance.

Sharing resources is an effective means of cutting costs, but since the robotics resources are physical resources, sharing across university sites creates problems of accessibility. For a robotics facility to be accessible we mean that:

- Getting 'into' the laboratory should incur minimal cost.
- Using the resources should incur minimal cost.
- Groups of students should be able to share and co-operate in the use of the resources.
- The resources should be 'relevant' to taught material.

The key constraint on sharing is, of course, the physical nature of the resources, since it is not convenient to travel to a remote site to use laboratory facilities. The Internet, however, provides a mechanism for overcoming this constraint [2]. It allows the laboratory to be transported to the user. This means not only transporting the robotics systems but also their environments. The large bandwidths and support for video compression and display that current networks and multimedia workstations provide make the transportation of these environments feasible. Thus, the creation of an easily accessible, shared robotics laboratory is now a realistic alternative to the severely limited local laboratory facilities that only a few institutions possess.

Netrolab is a funded project currently under way in the Computer Science Department at the University of Reading in collaboration with the Manufacturing Engineering and Operations Management Department of the University of Nottingham. The

---

*Accepted 31 December 1995

goal of Netrolab is the creation of a shared robotics laboratory on the Internet by 'networking' a set of physical robotics resources including vision and sonar sensing modules, a manipulator arm, and a mobile robot, all housed within an environment which we call Roboscape (ROBOtics landSCAPE) located at the University of Reading. It aims to allow the user to experience and understand robotics technology through a set of educational modules comprising teaching and experimental material, and to develop and pursue projects which investigate many aspects of robotics technology.

In this paper we present the Netrolab project and the Roboscape environment. We outline the hardware and software infrastructure we are putting together to support the configuration of the laboratory resources for teaching and experimentation. In the following section we will describe the requirements we are aiming to satisfy. We will then outline the architecture of Roboscape, including the physical environment and our software infrastructure. We will present a case study focusing on the use of the resources to support computer vision experiments and illustrate the concept of a virtual robotics teaching laboratory. In the final section we will draw our conclusions.

REQUIREMENTS

We identify five sets of requirements for the provision of a shared robotics laboratory for teaching, namely the coverage of the subject of robotics, support for experiments, for group-based experiments, for programming and finally for concurrent use of the experiment facilities.

Subject coverage

Robotics technology is generally covered from a number of different perspectives across a range of subjects, reflecting diverse interests [3–5]. For example, mechanical engineering is interested in the mechanical design of robots, electronic engineering is interested the electronics, interfacing and control of robots, and computer science is interested in programming models of intelligence, human interaction with the robots and multi-robot systems. We aim here not to draw boundaries, since interests can overlap significantly. Netrolab, however, aims to satisfy these diverse interests. Hence the collaboration of the Computer Science Department at the University of Reading and the Manufacturing Engineering and Operations Department at the University of Nottingham. The robotics and artificial intelligence courses taught in these two environments provided the basis for establishing the first sets of educational modules and for carrying out their validation.

Experiments

The conventional model of a laboratory environment comprises a set of resources which are configured as needed to support a diverse range of experiments. We are adopting a similar model for Netrolab. The primitive resources in this case comprise manipulators, grippers, sensors, cameras, mobile bases and programming tools including robot simulation packages. Experimental modules will be created by recruiting these resources as needed. In order to access these experiments across the Internet, the following additional requirements must be satisfied:

1. The physical robotic devices require hardware and software interfaces which are customized to support network-based access.
2. The resources must provide a set of services that can be used to configure a wide range of experiments.
3. Primitive and complex resources, the latter composed from primitive resources, need to communicate with each other and with the user, across the network.

Group experiments

Experiments can be performed either as individual or group work. Support for either mode of working is another requirement that must be satisfied. Group work can come in various forms, but we illustrate it here with the use of multiple controls for achieving co-operation in robotics tasks. We are specifically motivated by teleoperation, which requires the ability both to control a remote manipulator and to control the cameras used for viewing the remote environment [6]. Normally, one person controls the manipulator and another controls the viewing [7]. Netrolab will allow students to experience this type of co-operative working in an environment which is typical of many teleoperation environments in research centres around the world. An experimental module supporting co-operative experiments in teleoperation, for example, will allow the controls for the manipulator to be displayed on one workstation and those for the cameras to be displayed on another. These workstations may be located in the same or remote laboratories. This scenario is depicted in Fig. 1.

Programming

One further requirement needs to be satisfied in order to provide an effective support for project work, namely the ability for the student to access the software environment of Roboscape in order to build control architectures for the robots. Modelling and simulation software modules will be provided in order to support this programming model. Programs can thus be tested in the simulation environment and then integrated with the real environment. This will alleviate some of the pressure on the real resources. That is, while only a few students are accessing the real resources, many others will be able to use the programming modules to develop and test their programs in simulation.

Concurrent experiments

Finally, continuing the conventional laboratory metaphor introduced earlier, there must be support

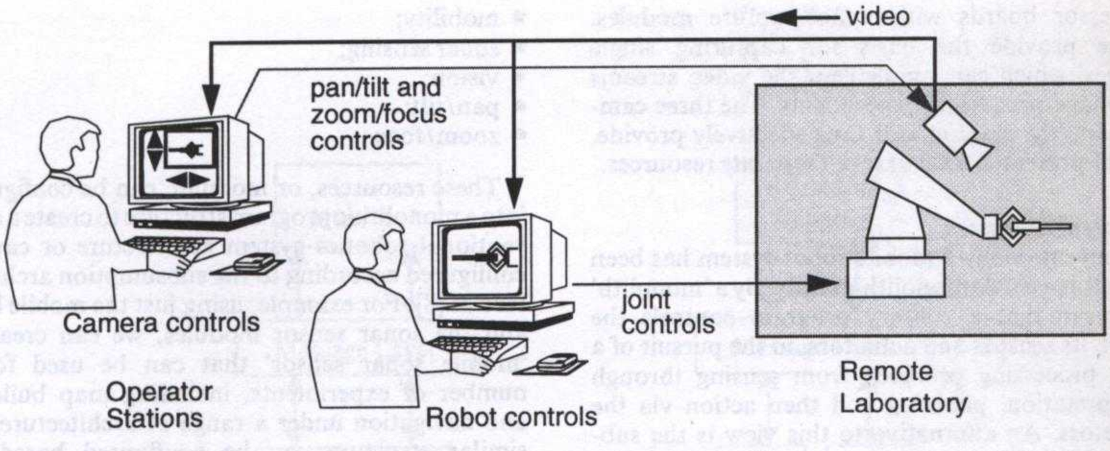


Fig. 1. Co-operative remote working in tele-operation.

for multiple experiments running concurrently. We term this 'laboratory fractionation'. Specifically, while one experiment may require a subset of the resources, multiple experiments may be possible if they do not share resources in common or only a limited set of resources are shared. This aspect of the service is to be investigated further as the Netrolab project proceeds, but our aim is to give the widest access possible by making the most efficient use of the resources.

SYSTEM ARCHITECTURE

Our starting point for defining the system architecture are the robotics resources. These are located within our laboratory, which we refer to as Roboscape, and include a robot manipulator and a mobile robot called Nero. We also have a set of three cameras with motorized zoom lenses mounted on pan/tilt heads for viewing the manipulator and the mobile robot as they operate within Roboscape. The basic architecture supporting the networking of these resources is illustrated in Fig. 2.

Hardware

The controllers for the three laboratory camera heads and the manipulator are served from a networked PC, running Linux (a version of Unix), via serial links. The video streams from the three cameras are captured using three real-time video capture boards resident on one of the laboratory Sun workstations. These boards can compress a video stream using a number of compression formats (e.g. JPEG and MPEG) and transmit it across the network. These provide the basic ability at present for viewing the laboratory. The video streams are also fed into a VME system housing three 68040

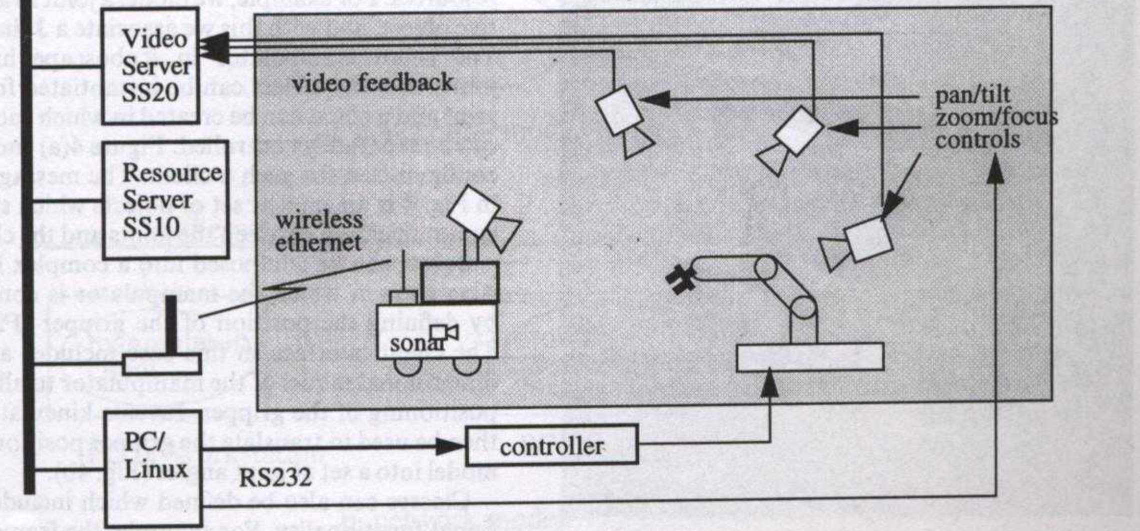


Fig. 2. The Roboscape environment.

processor boards with image-capture modules. These provide the basis for capturing single images, which can supplement the video streams for image-processing experiments. The three cameras and the manipulator thus effectively provide, for the present, a total of seven separate resources.

Robots as resources

Conventionally a mobile robot system has been viewed as a single monolithic entity. By a 'monolith' we mean that a solitary program controls the robot, its sensors and actuators, in the pursuit of a task; processing proceeds from sensing through interpretation, planning and then action via the actuators. An alternative to this view is the subsumption architecture proposed in [8], which introduces the idea of a set of decentralized processing modules running on a distributed architecture. The advantage of this architecture is that multiple sensor-to-actuator control loops can be active simultaneously.

We, in contrast, define the robot as a 'set of resources' which can be assembled together to achieve a task, but can be reconfigured in the light of new tasks and requirements [9]. In this conception the robot is not bound by the physical cohabitation of sensors and effectors on a single physical infrastructure. Nero, our mobile base (Fig. 3), thus comprises of the following resources:

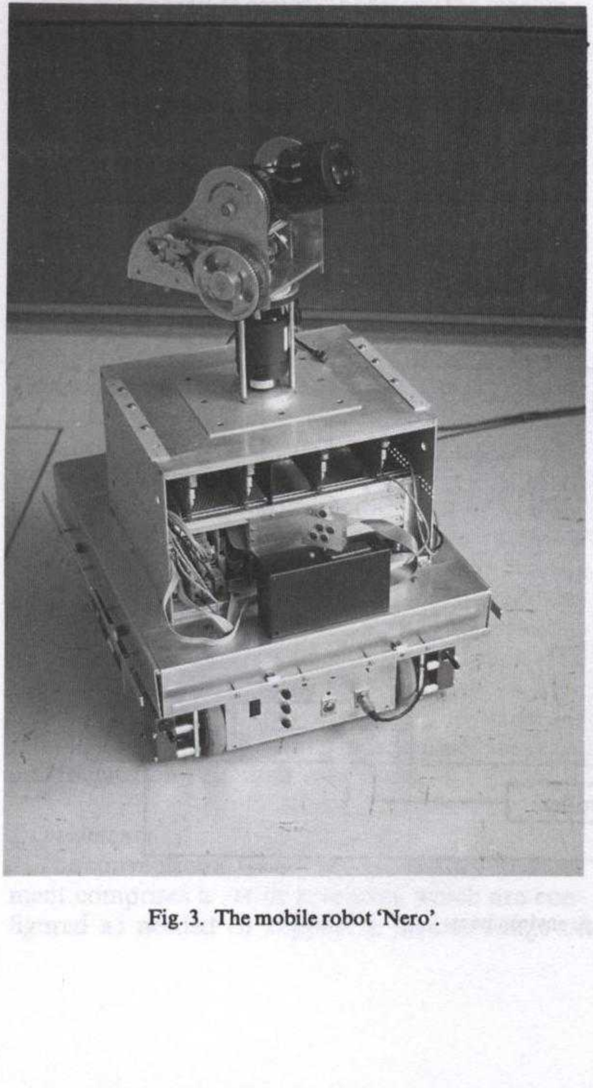


Fig. 3. The mobile robot 'Nero'.

- mobility;
- sonar sensing;
- vision;
- pan/tilt;
- zoom/focus.

These resources, or modules, can be configured into a monolithic program structure to create a conventional robotics system architecture or can be configured according to the subsumption architecture of [8]. For example, using just the mobile base and the sonar sensor modules, we can create a 'mobile sonar sensor' that can be used for a number of experiments, including map building and navigation under a range of architectures. A similar structure can be configured based on vision, where either the mobile base is stationary or can be moved about to create a 'mobile eye'. On the other hand, all three resources could be used to create a robot for experimenting with vision–sonar sensor fusion.

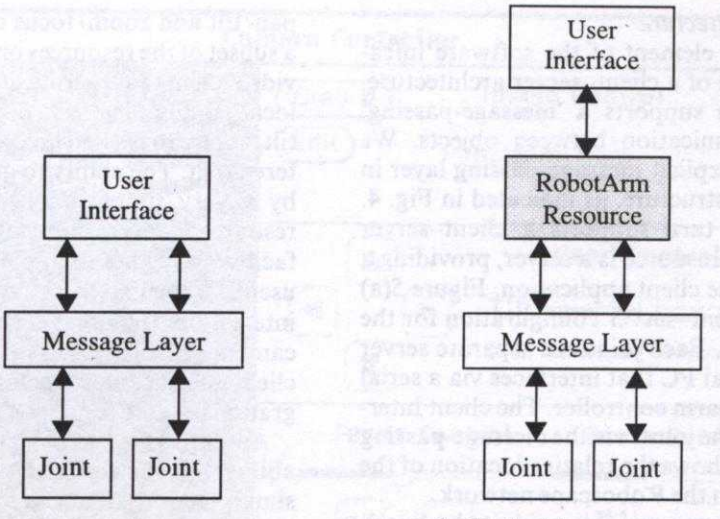
Software architecture

In order to exploit this resource-based model, we are implementing a software framework to support the networking of the robotics resources and to allow the flexible configuration of resources into educational modules. The framework is based on object-oriented techniques and is being implemented in C++. Object-oriented techniques provide a range of advantages for robotics applications as they do, for example, in window-based systems. Just as in the latter an interface can be constructed from a collection of resources, the widgets, so in the former an experiment can be configured from robotics resources. Interfaces to these experiments can in turn be constructed from the standard Windows widget set or from a library of specifically tailored interfaces.

All the advantages of reusability, modularity and extensibility associated with object orientation can be exploited to create hierarchies of resources and allow the relatively easy introduction of new resources. For example, we model a joint as a primitive object, and with this we associate a **Joint** class. The robot manipulator in Roboscape has five joints. A **Joint** object can be instantiated for each joint and a client can be created in which each joint can be separately controlled. Figure 4(a) shows the configuration for such a client. The message layer in Fig. 4 is an explicit set of objects which support communication between the joints and the clients.

Joints can be composed into a complex **Robot-Arm** class in which the manipulator is controlled by defining the position of the gripper (Fig. 4b). The client interface in this case includes a three-dimensional model of the manipulator to allow the positioning of the gripper. Inverse kinematics can then be used to translate the gripper position in the model into a set of joint angles (Fig. 4b).

Classes can also be defined which include additional functionality. For example, the frame grabber on Nero acts as a vision-sensing resource which

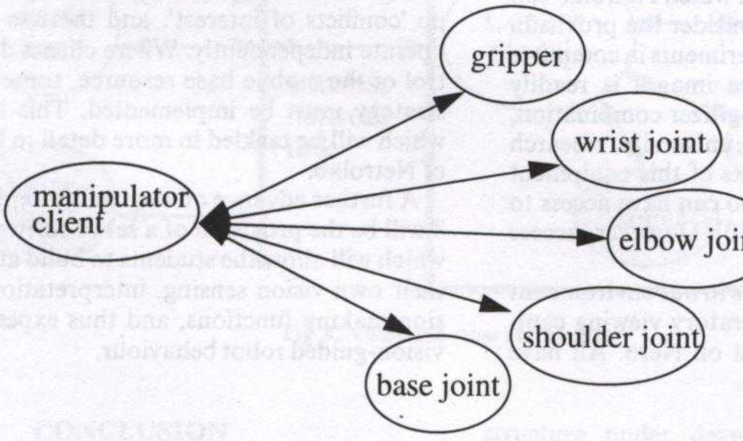


a. control of individual joints

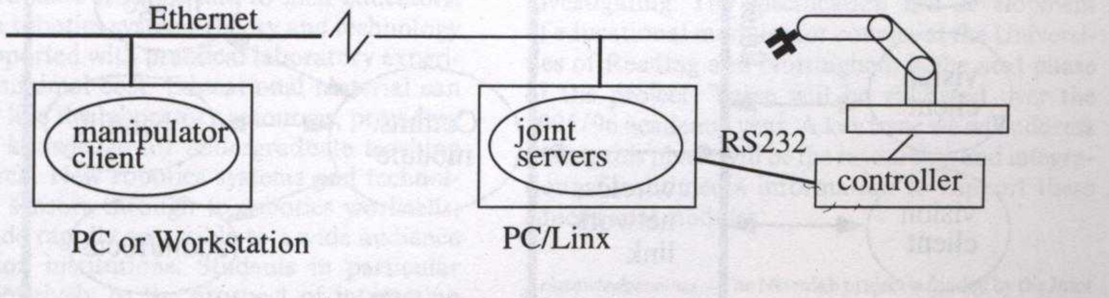
b. RobotArm resource

Fig. 4. Two forms of robot arm control.

serves images to clients. These images can be transmitted across the network and displayed on the client interface. Additional functionality can be added by introducing objects which mediate between the resource and the client to carry out image-processing and computer vision functions. A particularly useful function, for example, would be image compression, which would allow efficient use of the limited wireless bandwidth of < 1 Mbit/s between Nero, our mobile robot, and the main network. A complementary decompression module would also be required.



a. conceptual model



b. network model

Fig. 5. The manipulator client–server module.

Client–server architecture

The second key element of the software infrastructure is the use of a client–server architecture. Object orientation supports a 'message-passing' model for communication between objects. We have installed an explicit message-passing layer in our software infrastructure, as indicated in Fig. 4. This structure in turn supports a client–server architecture. Each resource is a server, providing a set of services to the client application. Figure 5(a) shows a simple client–server configuration for the manipulator client. Each joint is a separate server resident on the local PC that interfaces via a serial line with the robot arm controller. The client interfaces with each of the joints via the message-passing layer. Figure 5(b) shows the relative location of the server and clients in the Roboscape network.

If viewing resources are also exploited, experiments in the area of tele-operation can be carried out. If these two set of resources are provided as separate clients on separate workstations, one of the students can be control the robot manipulator and another can be control the cameras. This allows the students to experience experimental environments corresponding to those employed in advanced tele-operation research.

CASE STUDY

As a case study of the way in which Netrolab can have a role in teaching, we consider the provision of a simple tool to support experiments in computer vision. The ability to capture images is readily facilitated via a camera and digitizer combination, which is available in many university research environments. The falling costs of this equipment means that undergraduates too can have access to image-processing hardware [10]. However, access is still very limited.

The ability to capture images in our environment is supported by both the laboratory viewing cameras and the camera mounted on Nero. All have pan/tilt and zoom/focus controls. Figure 6 shows a subset of the resources on Nero configured to provide a camera targeting and image capture tool. A local communications module on Nero has pan/tilt, zoom/focus and image capture resources registered to it. The ability to grab images can be served by a very simple application incorporating one resource, namely the image capture resource. The facility we depict in Fig. 6, however, is much more useful, allowing the student to grab a region of interest in the image frame and to target the camera prior to doing so. Figure 7 shows a simple client interface constructed to demonstrate the integration of these functions.

A key feature of the Netrolab environment is the ability for multiple users to run an application simultaneously. Thus, a teacher can hold practical image-processing sessions within a computer workstation laboratory. Each student, including the teacher, can run the application on their own workstation.

Figure 8 illustrates a specialization of this scenario where different experiments are running concurrently. In this case a sonar sensing client is provided in addition to the vision client. The sonar resource on Nero integrates targeting and data capture. Thus, the student can direct the sonar to point in a particular direction, initiate a sonar pulse, and receive a digital representation of the returning signal. Since neither the vision nor the sonar clients in Fig. 8 have control of the mobile base, there are no 'conflicts of interest', and the two clients can operate independently. Where clients do seek control of the mobile base resource, some arbitrating strategy must be implemented. This is a subject which will be tackled in more detail in later phases of Netrolab.

A further advance on the facility depicted in Fig. 8 will be the provision of a set of software libraries which will allow the students to build and integrate their own vision sensing, interpretation and decision-making functions, and thus experiment with vision-guided robot behaviour.

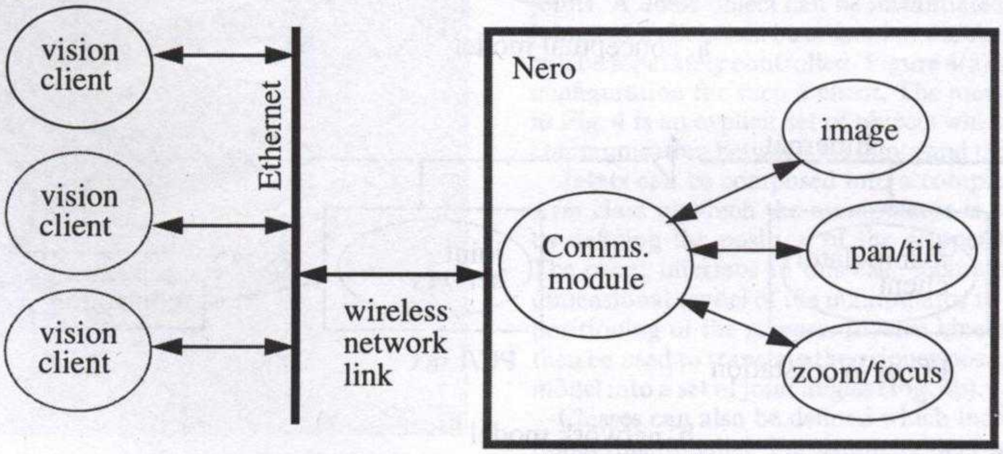


Fig. 6. Client application for image grabbing.

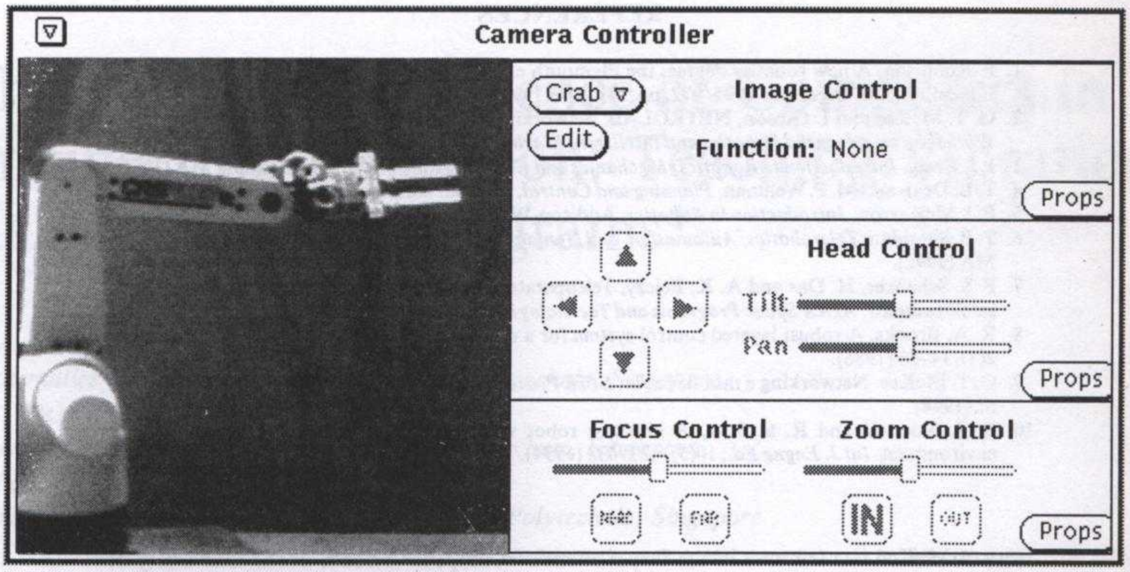


Fig. 7. Vision application interface.

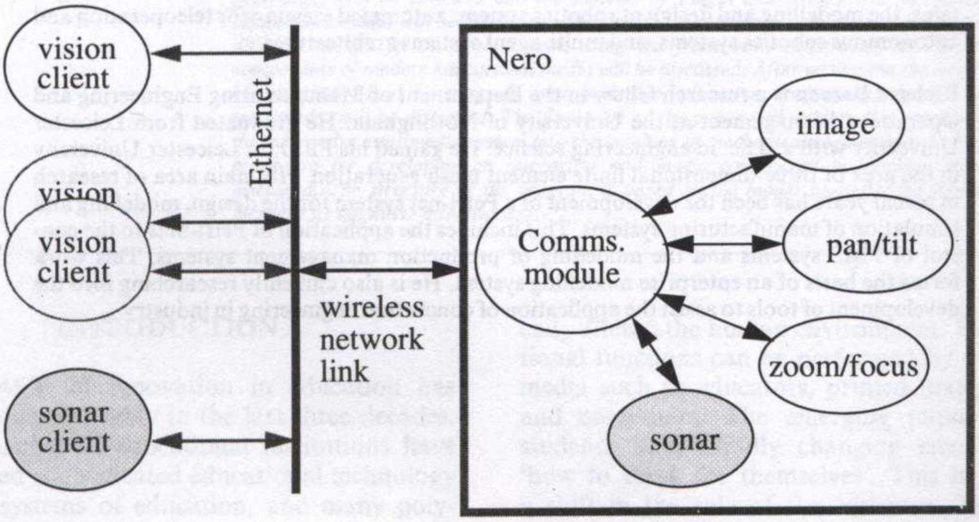


Fig. 8. Vision and sonar experiments.

CONCLUSION

The Internet and multimedia workstation technology can be exploited to bring robotics facilities to undergraduate students and to their educators. Courses in robotics systems theory and technology can be supported with practical laboratory experiments at minimal cost. Educational material can be shared, like the laboratory resources, providing a body of knowledge for undergraduate teaching and research. New robotics systems and technology, from sensors through to robotics workcells, can be made rapidly accessible to a wide audience in education institutions. Students in particular respond positively to the prospect of interacting with real systems.

In this paper we have presented the initial infrastructure under development within Netrolab. Extensive testing of this infrastructure is currently under way. Providing and controlling access to the resources is a major issue that we are currently investigating. The specification and development of educational modules for courses at the Universities of Reading and Nottingham is the next phase of the project. These will be validated over the 1995/96 academic year. A key issue we will address during this phase will be the resourcing and integration of multimedia information to support these educational modules.

Acknowledgements—The Netrolab project is funded by the Joint Information Systems Committee (JISC) of the Higher Education Funding Council for England (HEFCE) under its 'New Technology Initiative'.

REFERENCES

1. P. Robinson, A new robotics degree: the Plymouth experience, IEE Colloquium on Robotics and Education, *IEE Digest no.* 1995/072, pp. 1/1–1/8 (1995).
2. G. T. McKee and I. Gibson, NETROLAB: networking a robotics laboratory, *Proc. International Workshop on Advanced Robotics and Intelligent Machines*, Salford University (1994).
3. J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Addison-Wesley, Reading MA (1986).
4. T. L. Dean and M. P. Wellman, *Planning and Control*, Morgan Kaufmann (1991).
5. P. J. McKerrow, *Introduction to Robotics*, Addison-Wesley, Reading, MA (1991).
6. T. B. Sheridan, *Telerobotics, Automation, and Human Sypervisory Control*, MIT Press, Cambridge, MA (1992).
7. P. S. Schenker, H. Das and A. K. Bejczy, Teleoperator performance in simulated solar maximum satellite repair, *AIAA Space Programs and Technologies Conference*, Huntsville, AL (1992).
8. R. A. Brooks, A robust layered control system for a mobile robot, *IEEE J. Robot. Automat.*, **RA-**2(1), 14–23 (1986).
9. G. T. McKee. Networking a mobile robot, *SPIE Proceedings*, Vol. 2355, *Sensor Fusion VII*, pp. 55–61 (1994).
10. R. I. Noorani and R. S. Stits, A low-cost robot vision system to simulate a manufacturing environment. *Int J. Engng Ed.*, 10(5), 427–433 (1994).

Gerard McKee is a lecturer in the Department of Computer Science at the University of Reading. He graduated from the University of Manchester, Institute of Science and Technology (UMIST) with a B.Sc. (Hons) in electronics in 1980. He gained his Ph.D. at UMIST in the design of intelligent systems. He joined the University of Reading in 1987 where he founded the Active Robotics Laboratory in 1991 and is currently Director of the Active Robotics Research Group. His research interests are in the area of networked robotics systems, the modelling and design of robotics system, automated viewing for teleoperation and autonomous robotics systems, and multi-agent systems architectures.

Richard Barson is a research fellow in the Department of Manufacturing Engineering and Operations Management at the University of Nottingham. He graduated from Leicester University with a B.Sc. in engineering science. He gained his Ph.D. at Leicester University in the area of three-dimensional finite element mesh generation. His main area of research in recent years has been the development of a Petri-net system for the design, modelling and simulation of manufacturing systems. This includes the application of Petri-nets to the control of FMS systems and the modelling of production management systems. This work forms the basis of an enterprise modelling system. He is also currently researching into the development of tools to assist the application of concurrent engineering in industry.