Technical Program

15-20 November 1992
Hynes Convention Center
Boston, Massachusetts USA

SPIE's

OE/TECHNOLOGY '92

An Applications Symposium on Optics, Electro-Optics, and Lasers in Industry

OPTICAL TECHNOLOGIES

TOOLS FOR MANUFACTURING

INTELLIGENT ROBOTICS

Chair: David P. Casasent, Carnegie Mellon University

VISUAL COMMUNICATIONS AND IMAGE PROCESSING '92

Chair: Petros Maragos, Harvard University

Held in conjunction with

OPTCON '92
Helping you put Lasers and Optics to work

Educational Short Courses
Trade Show
Applications Workshops
Product Demonstrations

OPTCON '92 Sponsors

IEEE/Lasers and Electro-Optics Society (IEEE/LEOS)
Laser and Electro-Optics Manufacturers' Association (LEOMA)
Laser Institute of America (LIA)
Optical Society of America (OSA)
SPIE—The International Society for Optical Engineering

Sponsored by

SPIE—The International Society for Optical Engineering

Teleoperated mobile robotics instructional laboratory

Charles J. Cohen, Lynn Conway, Remzi Arpaci, and Alexander Ramos

University of Michigan
Department of Electrical Engineering and Computer Science
Ann Arbor, MI 48109

Abstract

Teaching students concepts about mobile and stationary robotics is difficult because there are few environments available for elegant and robust demonstrations. To aid student's understanding, robotic concepts need to be demonstrated not only in the lab but in the classroom. Since it is usually difficult to bring the equipment from the lab to the classroom, an environment for controlling and viewing the robots from the classroom is presented. The environment has several features: control from a distance, visually monitoring of actions, and feedback from different levels of system. A complete environment incorporating these features for teleoperated control of mobile robots and robot arms is detailed.

1. Introduction

During a robotics lecture, an instructor relies mainly on sketches, figures, and verbal explanations to present concepts. A laboratory section is used give students hands-on robotic experience. However, to make certain robotic concepts clearer to the students, it is beneficial for the instructor to control a robot during the lecture.

Unfortunately, the layout of classrooms makes such demonstrations difficult. Mobile robots need room to maneuver. Most robot arms are large and difficult to move. Additional equipment is usually required for controlling the robots. Special trips could be made to the laboratory, but that would be a waste of time and energy when only a minor point is being addressed.

Therefore, the need for remote viewing and control of the laboratory from the classroom is apparent [1]. This paper presents an environment specifically geared for classroom use.

Once the environment was created, it was apparent that not only were remote demonstrations possible, but the laboratory experiments themselves could be performed remotely. This idea is explored, and additional laboratory experiments are presented that illustrate concepts in telerobotics and remote monitoring.

The creation of this environment has shown us the need for an infrastructure in robotic architectures. No robot stands alone. As a computer is not just central processing unit, a robot is not just a mechanical link. A robot needs space to work, a computer for control, and feedback for measuring performance.

A robotic infrastructure enhances a scientist's ability to explore questions in areas such as kinematics, dynamics, path planning, and obstacle avoidance by providing the resources to study phenomena, monitor experiments, and record data. Multiple robot platforms can be access simultaneously by networking. Multiple views of the experiment are available from the video links. The data is sent back to a single location for easy recording and tabulation. A strong infrastructure provides the means for all these tasks.

In addition, a rich environment allows a wider variety of experimentation. Experiments on autonomy, coordinated action, artificial intelligence, and control are all easier when there are multiple agents available. What is needed is the infrastructure that brings different agents together in a simple, yet non-trivial manner.

The environment presented is only the beginning of the total infrastructure. In this paper we present the basic hardware and software structures needed for telerobotics and remote viewing. In Section II we examine the laboratory environment prior to the starting of this project. Section III details the project hardware specifications. Section IV details the project software specifications. Section V illustrates the possible telerobotic laboratory experiments that are possible with this environment. The last section discusses the future direction of this project and our conclusions.

2. The Laboratory Environment

The initial configuration of the robotics laboratory room is presented below. This laboratory provides students with access to robot arms, mobile robots, and computer facilities.

The eight robot arms stationed in the room have six degrees of freedom and a gripper. A personal computer is connected to each robot arm as a controller through the use of executable files or Pascal language programs.

The mobile robots available are the Cybermotion CARMEL [2] and the TRC Labmate. Both mobile robots have two degrees of freedom (drive and steer). Ultrasonics sensors are used by both for object detection, although CARMEL also has a vision sensor. CARMEL and the Labmate are controllable from a nearby Silicon Graphics Workstation (SGI).

The room has a connection to the video broadband - a type of cable television network that exists throughout a large portion of the University. The broadband consists of 99 video channels. A modulator is required to place a video signal on the broadband, one for each channel. Any number of cable ready televisions or VCRs can receive the video signal from a specific channel.

The University's computer network, known as CAEN, is connected to Internet, allowing electronic mail and file transfers with other networked computers. Initially the robot arm PCs were not connected to Internet, but the SGI's were.

3. System Hardware Design

Figure 1 shows the hardware configuration of the completed project.
The design allows teleoperational control of the robots, along with remote viewing of the environment. Because of the equipment used, it is easiest to have an SGI handle the robot control, and for a Macintosh to handle the video control. Refer to the next section for a discussion of the controlling user interface.

Control of the robot arms and mobile robots is transferred from the local computers to a single remote computer, a Silicon Graphics Workstation (SGI), which is connected to the computer network. The RTX PC's require an Ethernet board to connect them to the network. The SGI connected to the mobile robots is already networked, so no additional equipment is required.

Video transmission is achieved through the combination of video cameras, modulators, the broadband, and a demodulator. The video cameras are located in the laboratory environment. The video signal is sent to the modulator two ways: direct connection or signal transmission. The cameras attached to the RTX arms use a direct video cable connection. The cameras on the mobile robots use Rabbit transmitter/receiver boxes to send their video image to the modulators. Each pair of Rabbits is set to different frequencies (three are possible) so there is little interference between the signals. The Rabbits have a maximum range of about 90'. This is reduced to about 20' without line of sight.

Each modulator takes the video signal and pumps it onto a specific channel on the broadband network. Each modulator is set to a unique transmitting channel. The video out of the modulators is connected to the broadband.

A demodulator can takes the video signal from the broadband. Any cable ready television or VCR can act as a demodulator. The television can display the image immediately. A VCR can send the image to a TV or to a Macintosh screen via a Rasterop card (or any other video to Macintosh interface).

As explained in the next section, the Macintosh can also be used to control the VCR to tune to a specific broadband channel. A V-box is a device that allows a Macintosh to communicate with certain types of equipment, such as VCRs and camcorders.

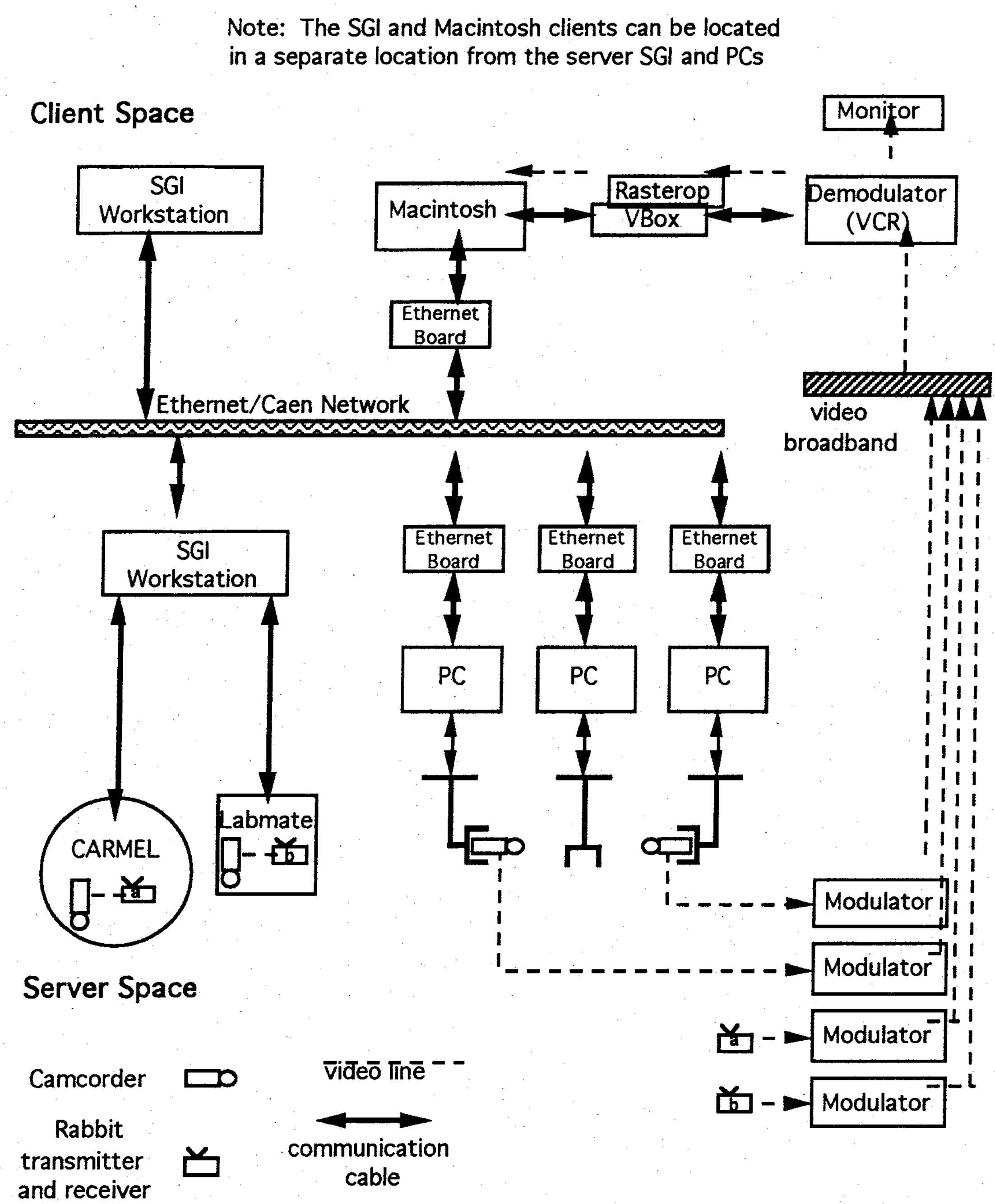

Figure 1 - Final Hardware Design

4. SYSTEM SOFTWARE CONFIGURATION

4.1 Robot Control

The Silicon Graphic's workstation at the top of figure 2 is the client machine, and controls the robot arms and the mobile robots. The C language interface is graphical to allow for complete and easy control of the platforms. Because the interface also uses X-windows, any machine that can support X-windows can also run this program.

The interface is divided into three main windows: a control window, a text log window, and a map window. The control window is the primary window for the interface, and is described in more detail below. The text log window keeps a file containing a time coded list of every command performed. This file is used as an aid to evaluate the interface or as part of a students laboratory report. The map window displays an ultrasonic sonar map based on data received from a mobile robot. This data can be stored, retrieved, or filtered as desired.

The commands for everything from communication to robot motion to data manipulation are executed through the control window. These commands are accessed through pull down menus or through toggle switches, depending on their functions. The main sections are summarized in the following paragraphs.

Communications: Initiates and terminates communication with the remote robots. Any number of communication links can be open simultaneously, but only the active port will send and receive data.

Robot Control: This menu contains the specific commands for controlling a robot. For example, a mobile robot can be commanded to move forward, or a robot arm can be commanded to open its gripper. The variety of commands is dependent on the robot being controlled. Of course, some commands cause the robot to send back information, such as current position or a sonar map.

Goodies: Contains extra commands such as defining communication macros, opening an x-window clock, and opening a calculator.

Script: This useful tool allows users, even those not versed in C, to program the robots to perform complex tasks. This command initiates a powerful scripting feature for assembling the previously discussed robot commands in an intuitive C programming language type manner. Command looping, conditional clauses, defining variables, and creating equations are also supported. The commands are invoked by translating the script into C, compiling, and executing them. All of these actions are hidden from the user.

VFH bulletin: This command initiates the virtual field histogram obstacle avoidance method [3] on the active mobile robot. While the robot is in VFH motion, no other commands to that platform are possible.

Joystick: This command allows joystick control of a mobile robot platform.

Sense: This command displays the distance measurements from one sonar sweep from a mobile robot platform in a graphical format for easy comprehension.

RTX bulletin: This command opens three windows representing two dimensional views of the current RTX robot configuration, as viewed along the x-axis, y-axis, and z-axis (figure 2). This allows a three dimensional view and control of the robot arm on a two dimensional screen. Each joint can be controlled either individually by using a slider connected to each window (forward kinematics) or by selecting the final end effectors position by clicking on any of the three viewing windows (inverse kinematics). This allows a more intuitive type of control than is provided in the Robot Control Menu.

Help: A menu for getting help on the various keys and commands.

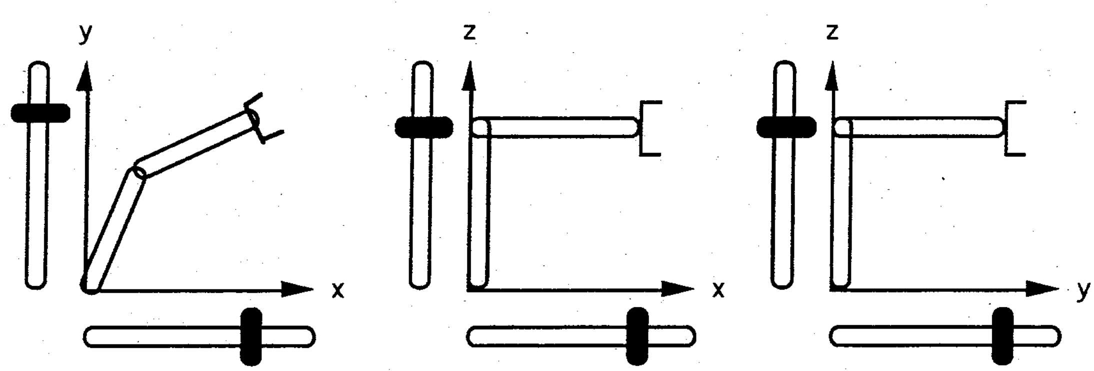

Figure 2 - Robot Arm Windows

4.2 Video Control

As explained in the hardware section, the cameras viewing the robots have their images on various channels on the broadband network. A Macintosh computer on the client control end can command a VCR to switch through the various channels on the broadband, allowing the viewing of all the camera images. The system requires the VISCA driver package to send commands from the Macintosh through the V-box to the VCR (see figure 1).

Currently, the system allows switching between any channels on the broadband, allowing the user to view each camera input at will. However, since the Macintosh is connected to the CAEN network, it is possible to send data back and forth to the SGI. The SGI knows the locations of the cameras mounted on the robots. The Macintosh controls the camera being viewed. The combined information and control ability allows coordinated control and viewing. This is further described in the experiments section.

4.3 Server Control

CARMEL, the Labmate, and the RTXs are the robots that are controlled at the server end. The controlling software is detailed in the following paragraphs. In general, the high level control is implemented at the client (SGI) end, while the basic motions and actions are handled at the server end.

CARMEL has it own on board PC computer, which receives commands from the SGI to run the ultrasonic sensors, vision system, and drive motors. Sonar readings, a vision map, and position coordinates are sent back to the SGI upon request.

The Labmate has the same structure as CARMEL, except there is no controlling computer that can be programmed, just low level drive and position controls. Also, the Labmate does not have any onboard vision processing, although a camera can be added to transmit live video via a rabbit to the broadband.

The RTX arms are controlled by their connected PC computer. They receive position commands and sends position information (x, y, z, yaw, pitch, and roll) upon request for the SGI. The position of a robot arm can be detailed in Cartesian, joint, or motor coordinates.

5. SAMPLE ROBOTICS LABORATORY EXPERIMENTS

This section illustrates example laboratory experiments made possible with this telerobotic environment. Control from a single client allows for easier coordinate action between the various robot platforms. The video displays the general configuration of the system. The graphical interface shows the exact configuration of each of the robot arms. Because the devices are connected at a single client, experiments in coordinated actions, such as tracking, is possible. Figures are provided to illustrate the basic concept behind each experiment.

Only the experiment's general overview is given. During all experiments, the RTX arms with the cameras should be active so that viewing from different locations is possible.

5.1. Obstacle Avoidance

The student controls a mobile robot in an environment with walls and obstacles. The student tries to maneuver the robot from one location to another by hand (i.e. using move forward and rotate commands) while avoiding objects that are seen with the sonar sensors. Other students set up the environment, if desired.

After the students have practiced maneuvering the mobile robot through the environment, they are to use the scripting feature and create an automatic obstacle avoidance program. This can be a very simple program. For example, the mobile robot can just turn left when an object is sensed in front, and so on. The students can race to see which reaches the goal first without colliding with an object.

This experiment will introduce students to basic obstacle avoidance techniques and real world problems such as sensor noise and path planning.

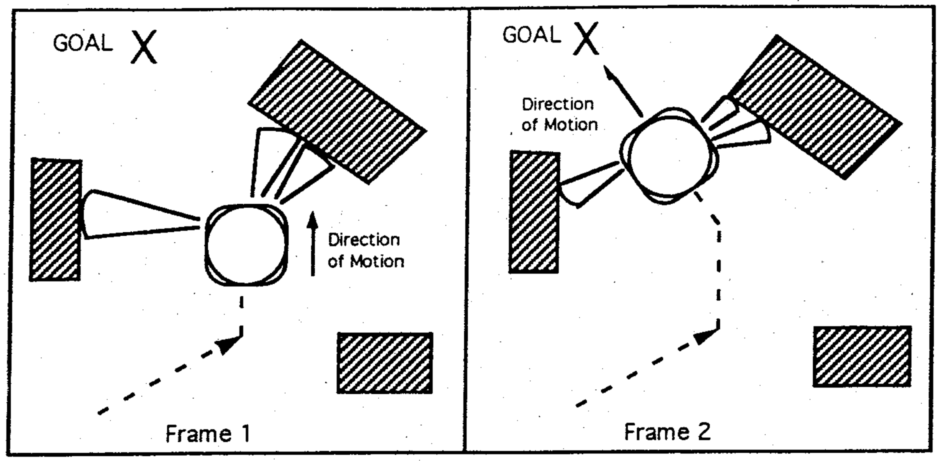

Figure 3 - Obstacle Avoidance

5.2 Manual Construction

While viewing the laboratory through the live video feed, the student uses the robot arm to stack blocks. The robot arm holding the camera is also allowed to move. This teaches the student about assembly and basic vision concepts.

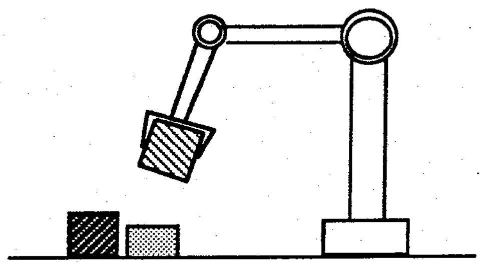

Figure 4 - Manual Construction

5.3 Pallitize

In this experiment, the student will use the robot arm to fill a pallet. The pallet's dimensions and orientations are left as variables. The robot arm has to grasp bocks from a set feeder position. This introduces basic ideas of the geometry needed for robot arm tasks.

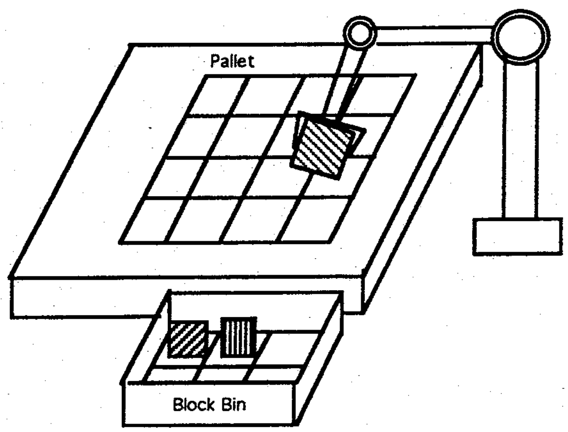

Figure 5 - Pallitize

5.4 Coordinated Pallitization.

As an extension to experiment 5.3 above, the student programs one of the robot arms with the camera to track the main gripper that is pallitizing. Again, this is a geometric lesson, but also some planning is involved. When the gripper moves out of the camera's field of view, the arm with the camera should track the gripping arm.

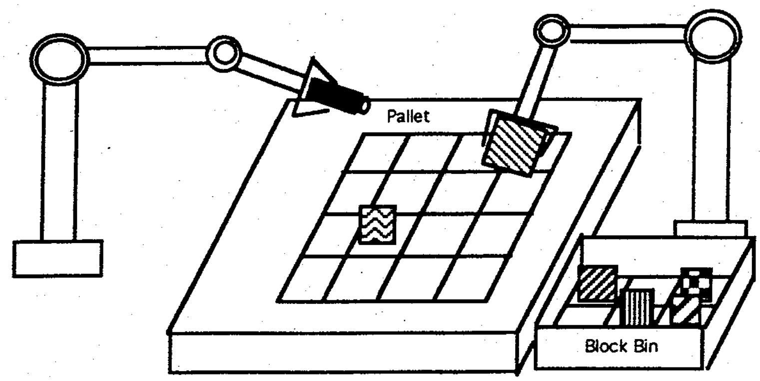

Figure 6 - Coordinated Pallitization

5.5 Coordinated Motion and Viewing.

A variation on experiments 5.1 and 5.4 is to have the robot arm track the mobile robot. The absolute positions of the mobile robot are known, and these can be mapped on to the viewing field of the camera, which is controlled by the robot. Again, a question of how much the arm needs to be moved must be answered.

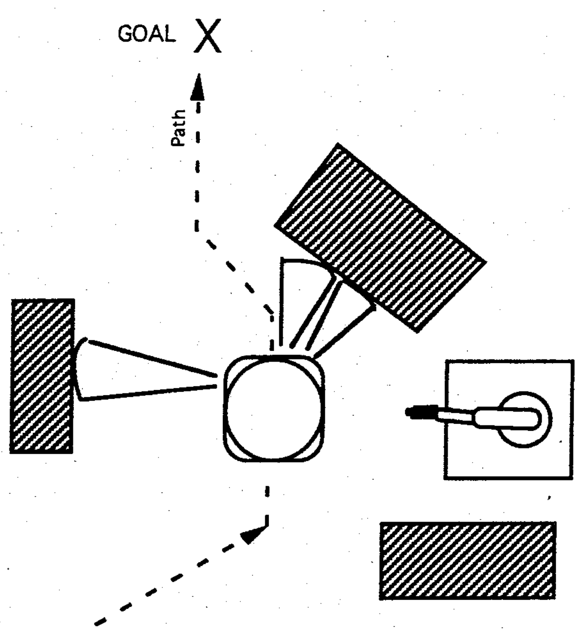

Figure 7 - Coordinated Motion and Viewing

5.6. Catch and Carry

Two RTX platforms have a number of objects that must be loaded onto the two mobile robots and delivered to various sites. The object of this lab is for the students to write a routine that will plan which robot arms load the various mobile robots (the order and the timing) to most efficiently get all the objects delivered in the shortest amount of time. A variation of this lab would be to have certain items that had to be delivered in a time critical fashion. This type of lab is an introduction to thinking about artificial intelligence techniques.

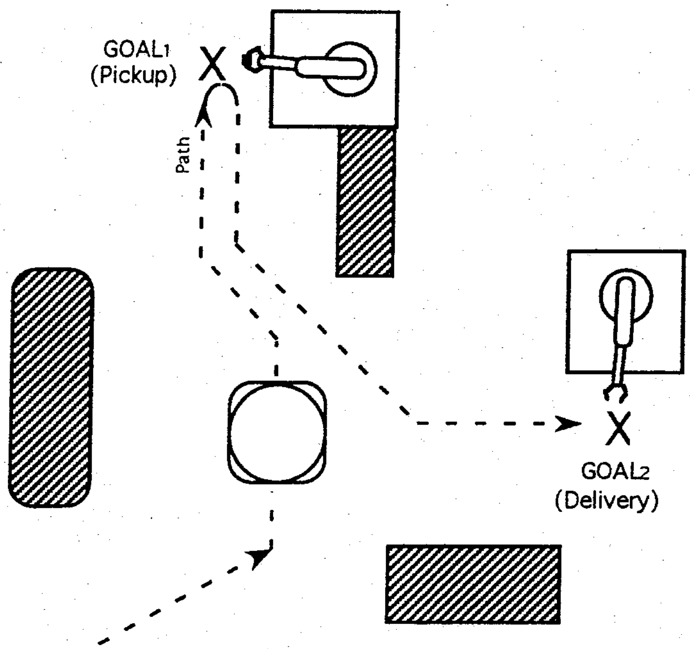

Figure 8 - Catch and Carry

6. FUTURE WORK AND CONCLUSIONS

The system described in this paper presents the groundwork for the infrastructure that we are evolving for a telerobotic environment. One future goal of this project is to make the entire robot arm laboratory, which is part of the Introduction to Robotics course, fully telerobotic. To achieve this goal the scripting feature is used. Experiments such as the following [4] will be easily possible exploiting telerobotic lab access:

1. Forward Kinematics - using just the motor coordinates provided to create smooth elegant motion via joint coordinates.
2. Inverse Kinematics - given the Cartesian coordinates, compute the required joint angles to achieve the desired end effector position.
3. Straight line motion - move from one point to another in a straight line interpolation.
4. PID control - test the effects of changing the gains on the proportional, integral, and derivative feedback controls.

Another step for this environment is to add additional types of robot arms and mobile robot platforms. Given more mobile robot platforms, the area of coordinated search and exploration can be researched, as can other general mobile robotic tasks. Small robot "necks" which pan and tilt a mounted camera are useful to have to better monitor an environment.

Because of the presence of video feedback, pattern recognition capabilities could be added. Such a vision system could identify objects. Students could experiment with the techniques involved with identifying and manipulating objects acquired from the vision sensors.

Currently, the client program run on the SGI is portable to any machine with X-windows. Ultimately, because Macintosh computers are more readily available than SGI workstations, it is desired to port the entire system (and not just VCR control) on to the Macintosh computers using their windowing system. This would allow a more user friendly environment for the students to use, while also reducing the number of machines for this system to one single controller.

We have presented an environment that enables remote viewing and control of a robotic laboratory. Multiple robot platforms are controlled and various experiments have been illustrated. Students and scientists can easily add their research into this pervasive environment, giving them the ability explore ideas and record results.

7. References

1. L. Conway, R. A. Volz, M. W. Walker, "Teleautonomous Systems: Projecting and Coordinating Intelligent Action at a Distance," *IEEE Trans. on Robots and Automation*, Vol 6, No. 2, pp 146-158, April 1990.

2. D. Kortenkamp, C. J. Cohen, M. Huber, et.al, "Integrating obstacle avoidance, global path planning, visual cue detection, and landmark triangulation in a mobile robot," in this volume.

2. J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol 19, No. 5, pp. 1179-1187, October 1989.

3. Y. Koren,Robotics for Engineers, McGraw-Hill, New York, 1985.