International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN : 2278-800X, www.ijerd.com
Volume 4, Issue 11 (November 2012), PP. 47-50

Internet Controlled Robots

Anil Kumar Malaga[1], Ramesh Ch[2]
[1,2]*Assistant Professor, Department of Mechanical Engineering, PACE Institute of Technology & Sciences, Ongole.*

Abstract:-This paper describes an exploration in Internet-based control of robots. The objective of this work is that Internet communications can be exploited to achieve greater productivity from machines with local intelligence. Local intelligent systems contact human experts to solicit advice when a problem facing the machine is beyond its cognitive capabilities. This topic is explored in the context of a robot performing a sample domestic task (ex: sorting laundry). A system is to be designed that has limited autonomous competence, but which proves to be significantly more productive through the use of occasional Internet-based human supervision.

I. Introduction

The Internet creates many technological opportunities, one of which is the ability to use a standard network infrastructure to control robots from a remote location. Internet-controlled robots will have valuable applications in automation and, more futuristically, in space and terrestrial exploration and in home robotics. Much research has focused on using Internet connectivity to control robots, and Internet robotics can be regarded as the natural extension of research in remote control. Previous work in Internet robotics demonstrates that:

1. It is possible to control robots over the Internet. However, this technology is not yet mature due to long and unpredictable delays, especially on heavily loaded IP networks that lack any provisioning for Quality-of-Service.
2. There is some evidence that it is possible to implement remote non-real-time robot control.

This paper offers an exploration of the potential for such future applications of Internet-based robotics. In this approach, local intelligent systems contact human experts to solicit advice when a problem facing the machine is beyond its cognitive capabilities. Simple tasks may be performed autonomously. For example, it is easy to write a program that directs a robot to go to point A, pick up an item, go to point B, and drop the item. If the location of the item were always the same, the robot would continue to repeat the task indefinitely and would never get tired of doing the same procedure. A more complicated scenario would be more difficult to automate. For example, if the system is required to perform different actions depending on the item color, the robot will start making mistakes. The developer will also need to develop and trouble-shoot the vision analysis code. While intelligent systems are getting more sophisticated, achieving adequate competence for autonomous operation in complex environments is not yet feasible. However, partial solutions may still be usable if augmented with occasional human assistance. With human-assistant capability, a robot may ask for help when it is confused. The idea is to allow the robot to complete the system sub-procedures on its own and ask for human help to overcome the more difficult ones. The approach has clear applications to automation. For example, an automated manufacturing plant may be relatively predictable and robustly controlled, but occasional diagnostics, corrections, and resets may be required. These operations might be performed via the Internet, possibly collaboratively, by expert supervisors located anywhere in the world. Furthermore, the approach can enable innovative applications, such as lumbering, mining, space exploration, and domestic service operations through Internet-based human supervision.

II. System Overview

The system consist a robotic arm, two cameras, a PC controller, and a Web server.

A. Robot & controller

The robot had been retrofit for open-architecture control. While the robot had limited workspace, payload, speed, and precision, but had safety, which is a significant consideration in remote control. The robot had five degrees of freedom in a serial kinematic chain, similar to popular industrial designs (see Fig 1).

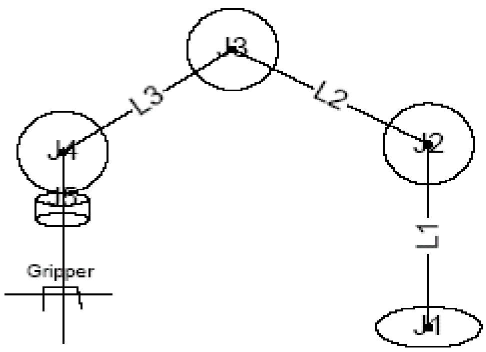

Fig.1: Robot Kinematics

The robot was interfaced at the torque level to an analog output board within a PC control computer. Incremental encoders on the joints were interfaced to encoder counters within the I/O card hosted by the PC.

B. Actuators & Sensors

The actuators consist of the joint motors and the robot's gripper. The sensors included motor encoders, an optical distance sensor, a gripper mounted black-and-white camera, and a color Web camera.

C. *Operating System*

The QNX operating system, a real-time operating system derived from UNIX is required. QNX supports real-time multi-tasking; i.e., multiple processes can be prioritized and scheduled to run independently, emulating parallel execution. Multi-tasking is required to run concurrently several control and communication processes. Communications among the separate processes is coordinated through semaphores. Semaphores were used to drive processes at fixed rates and as a mutual exclusion mechanism.

D. Web Server

The Apache Web server is used to run as the front-end for user interaction. In turn, the Web server communicated with the QNX controller and relayed user commands to the robot. The NT machine connected to the robot controller through Ethernet. In general, the Web server and the robot controller could be installed on the same or on different computers. A single-machine installation is characterized by reduced hardware requirements and by fast communication between the Web server and the controller. The separation of front- and back-end hosts can result in legacy with existing platforms and can lead to higher system scalability and security.

E. CGI (Common Gateway Interface) scripts

The Web server initiated robot operation by invoking a CGI script that ran at the Web server side (see figure 2).

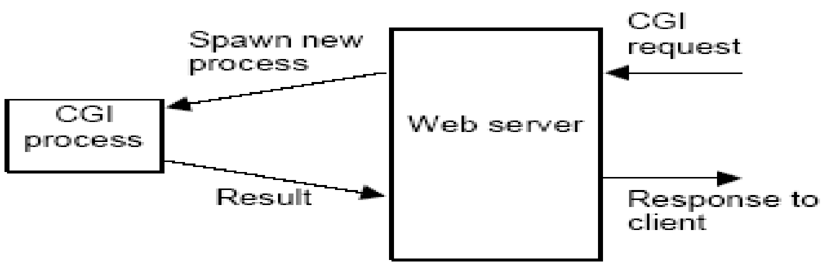

Fig.2: Common Gateway interface

In turn, the script embedded TCP connectivity to relay instructions to the robot. Furthermore, the scripts gathered feedback from the robot and passed it on to the user. As a result, CGI scripts provided a module to communicate and pass information back and forth between the server and the controller. CGI is an acronym that stands for "Common Gateway Interface" and refers to the protocol used to pass arguments from the Web server to the script. The Apache server had a handler for running CGI scripts as threads that are part of the Web server process: when the server invoked a CGI script, a new thread must be created within the Web server process to execute the script.

F. Vision Analysis

A Matrox card can be used as the image frame grabber. The Matrox card has a driver for Windows NT but no driver for QNX. Therefore, the Matrox card was installed on the same NT machine that hosted the Web server. The NT machine also performs vision analysis. Thus, images from the frame grabber were sent to two processes: a CGI script within the server process to forward the images to the end user and a C process that analyzed the images (see figure 3).

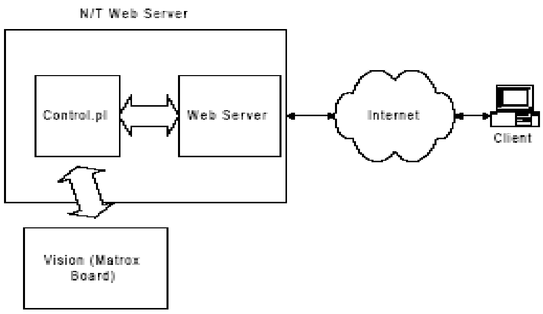

Fig.3: Client Server Architecture

The Matrox card is to be connected to a gray-scale camera, which was mounted on the robot's gripper. The camera and frame grabber provided 512 x 480 frames with pixel intensities in the 0 to 255 range.

G. User Interface (UI) presentation

The Web interface can be written in HTML. The interface had feedback buttons for a user to transmit commands to the Web. The commands would be relayed to the robot controller for execution. The frame grabber took a snapshot of the robot's view and the Web server encoded the saved image as part of the HTML display. The image was a black and white JPEG picture and it projected the robot's point of view. Such pictures were used as an input means. The user could click on the actual picture on the Internet browser, and this action would invoke recording the actual pixel X and Y coordinates. These coordinates were then delivered to the Web server and a CGI script relayed the data from the web server to the robot controller. Image-space coordinates were translated into robot-space coordinates, transparent to the operator, to command the robot to move to the selected location.

H. Web Cam

The buttons and the servo control were the input elements; however the user also needed feedback from the system. There were two forms of feedback: images from the robot's viewpoint (with the gripper mounted camera) and a wider-angle side view from a Web camera (a Webcam). The first camera was mounted on the robot's arm and its frame was refreshed only after the robot arm completed a movement. A second camera was mounted near the robot with a side view of the robot and its workspace and provided a real-time view of the robot environment.
To display the video stream, the client can use a Java applet, i.e., a Java byte code executable that was dynamically downloaded into a browser over the Internet. The applet continuously downloaded video streams from Webcam.

I. Client site

The human supervisor interacted exclusively with the Apache Web server and with the webcam video server. In practice, the operator could use any of the commercially available Web browsers to supervise the robot and download video streams. When the user clicked on a button or selected an option from a drop-down menu, the user choice was sent to the Web server in a standard HTTP request. The server would then trigger a CGI script that communicated with the robot controller through a TCP connection. Meanwhile, the webcam server would continuously push video frames to the Java applet running on a Java Virtual Machine within the browser, presenting the video feed to the user. The client site can use such as Java-enabled Web browsers, so that the human interface to the robot required no specialized software or hardware beyond what is already commonly available.

III. Human / Machine interaction

Internet-based control makes it possible for a remote user to direct robot operations from a remote location. However, network connectivity is inherently subject to time delays, which constrains an interface designer to limit the communication demands between the two hosts. Direct teleoperation, as described in, requires low latency to be safe and effective. In contrast, if most of the control communication is in the form of low-level commands, such as unidirectional movements, these can be interpreted and expanded into real-time execution with only low-level intelligence local to the robot. If robots can handle simple local tasks as well as basic survival skills, such as obstacle avoidance, the latency requirements would be greatly reduced. The principle of supervisory control motivates our adoption of coarse-grained interfaces such as drop down menus and point and click interaction. As for control feedback, it should provide reliable real time information about the robot's motion to a remote supervisor, and motivates our use of a streaming video server at the robot site and of the corresponding client applet at the user site.

IV. Imaginary Example

Fig.4: Imaginary Example

Let us take three baskets with three types of washcloths: bleachable, non bleachable, and ambiguous. The robot must be preprogrammed to pick-up the washcloths from a source basket and then use its cameras and visual analysis program to determine the color of the washcloth. The robot would then drop the washcloth into either the bleachable basket or the non-bleachable basket, depending on the analysis result. A pick-and-drop sequence was considered a "task". The entire sorting procedure was a group of tasks. If the robot had a problem determining the color of a washcloth, it should pause and ask for help from the user, who communicated with the robot through an Internet browser. The user could assist the robot by using buttons from the Web page. The human advisor looked at the picture that was shown on the browser and clicked on the corresponding radio button on the panel to provide guidance. The robot then completed the task according to the human interaction then continued with the remaining tasks on its own. In the experiment above, the "human assistance" system helps the robot to be more effective when performing difficult tasks.

V. Discussion and Conclusions

In this paper, a system is discussed with the concept of internet-controlled robotics. This systems offer a new concept to automation. Building a fully automated, reliable system can be infeasible or cost prohibitive for complex tasks using current technology. Human assistance can lower development cost significantly, and the resulting system can begin to be operational and productive sooner than a fully autonomous system. With human assistance, emerging technologies could be utilized more rapidly, since absolute dependability is not required. Thus, with human assistance, imperfect technology could be productively deployed in demanding scenarios. Immediate applications would include remote exploration, remote experimentation, hazardous environment operations and defense applications. More futuristically, this approach could help lower the cost of achieving adequate competence in domestic robots. Besides lowering the cost of building a system, our demonstration has illustrated that human assistance can improve the consistency of a procedure. From the laundry-sorting task example, the system performed the operation error free with the help of human supervision. Current and future developments are inspired by the need to add reliability and consistency to complex robotic systems. To this end, we have tried to implement a sophisticated software platform that is based on mobile intelligent agents and on local controls and low-level intelligence for performing force controlled contact operations.

References

[1]. Craig, J.J. *"Introduction to Robotics, Mechanics and Control,"* Addison Wesley.

[2]. "*Sensor Based Planning for a Rod Robot*", H. Choset and J. Burdick, 1996 IEEE Int. Conf. on Robotics and Automation.

[3]. G.S. Chirikjian and J.W. Burdick, *``A Hyper-Redundant Manipulator,"* IEEE Robotics and Automation Magazine, December 1995, pp. 22-29.

[4]. "*Controllability of Kinematic Control Systems on Stratified Configuration Spaces"* ,Bill Goodwine and Joel Burdick ,IEEE Trans. on Automatic Control, vol. 46, no. 3, pp. 358-368, March, 2001.

[5]. *"Determining Task Optimal Modular Robot Assemmbly Configurations"*, I.M. Chen and J. Burdick, Proc. IEEE Int. Conf on Robotics and Automation, 1995, Nagoya, Japan.

[6]. *"Kinematic Reducibility of Multiple Model Systems"*, T. Murphey and J.W. Burdick 2003 IEEE Conference on Decision and Control.

[7]. *"*Vaughan, R. N. Sumpter, A. Frost, and S. Cameron. "Experiments in Automatic Flock Control." Edinburgh, UK, 1998.

[8]. D. Michie, *"Machine Learning in the Next Five Years,"* Proc. of the Third European Working Session on Learning, Glasgow, 1988, pp. 107-122.

[9]. T. Inamura, M. Inaba and H. Inoue, *"Acquisition of Probabilistic Behavior Decision Model Based on the Interactive Teaching Method,"* Proc. of the 9th International Conference on Advanced Robotics, Tokyo, 1999, pp. 523-528.

[10]. Mark Yim, David G. Duff, and Kimon D. Roufas. *"PolyBot: a Modular Reconfigurable Robot."* IEEE International Conference on Robotics & Automation. April 2000.