Microrobot teleoperation through WWW

András Lassó[1], Tamás Urbancsek[2], Ádám Helybély[3]

[1]PhD student, lasso@seeger.iit.bme.hu; [2]graduate student, urbi@topcat.iit.bme.hu;
[3]graduate student, hadam@topcat.iit.bme.hu
Budapest University of Technology and Economics
Department of Control Engineering and Information Technology
H-1117 Budapest, Pázmány Péter sétány 1/D, Hungary

ABSTRACT

Application of robotic devices can be significantly extended by teleoperation – controlling them from a remote place. Internet, considering its wide spread and accessibility, is and ideal media for data transmission between a robot and its operator, but the limitation in bandwidth, the notable delay, and the unreliability of the network introduces several problems.
A teleoperated mobile microrobot system is presented in the paper. The operator has real-time remote access to the microrobot, through a web browser running on any computer connected to the Internet.
The low-level control of the robot is provided by a robot controller board and a video camera with an on-board digital signal processor. These devices controlled by a a PC connected to the Internet, which compress and transmits the data using real-time supporting transport protocol towards the operator's Java applet. The operator can formulate commands for the microrobot and observe their proper execution through the graphical user interface of the applet.

I. OVERVIEW OF INTERNET TELEROBOTICS

A. Internet telerobotics

As Internet became a standard medium for digital communication more and more systems used it for exchanging data between a device and its controller.

Though Internet has unique strengths (relatively low-cost, wide-spread), the transmission has the following limitations:

- throughput: limited bandwidth, depending on network congestion
- delay: time-varying (delay jitter)
- packet loss: routers may drop packets in case of heavy network traffic.

These limitations impose significant problems in application of Internet for real-time teleoperation of robotic devices.

Including the Internet in the closed control loop would cause instability and strict speed limitations should be applied (e.g. following a “move-and-wait” strategy). This problem is addressed by the human supervisory control architecture ([1]) There are efforts to apply traditional control theory principles to packed-switched systems by statistically modeling the network and compensating its imperfections [2].

Visual feedback plays an extremely important role in telerobotic applications, and delivering it to the user requires significant bandwidth, therefore its transmission has to be investigated with distinguished attention. There are numerous protocols are developed already that support real-time streaming media transmission via Internet (e.g. by inserting timing information, defining QoS requirements, and implementing services that use these information) [3, 4], and others emerging for general purpose and special applications.

B. *Web telerobotics*

The world wide web is one of the most popular service of the Internet, it is easily accessible from almost everywhere in the world, and it is also very rich in term of graphical presentation, communication methods and interactivity; thus it can serve as a platform for teleoperation very well.

Communicating with a remote machine through the WWW is possible using HTML forms and image maps using the common gateway interface (CGI) (e.g. the Mercury Project, 1993; Australia’s telerobot). However the set-and-submit cycle mechanism of CGI provides only very limited interactivity, and nowadays it is sparcely used int teleoperation systems.

Java-based solutions provide more interactivity (instant user feedback is possible), sophisticated user interfaces (easy-to-use graphical user interfaces, support for augmented and virtual reality) and improved communication architecture (stream or datagram based sockets, advanced communication frameworks for special applications) (e.g. PumaPaint; Web Interface for Telescience project – WITS).

C. *Real-time transport protocol (RTP)*

In teleoperated systems observing and controlling the remote device is performed in real-time. Internet today can not provide services for hard real-time operations, so Internet teleoperation works with soft real-time constraints (the devices are are teleoperated using the human supervisory control architecture).

The Network Working Group designed a protocol that provides end-to-end network transport functions suitable for applications transmitting real-time data: the real-time transport protocol (RTP [5]). RTP is designed primarily for audio and video data transmission in mind but it is also applicable for other data types used in teleoperation.

RTP does not address resource reservation and does not guarantee quality-of-service for real-time services. The data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery in a manner scalable to large multicast networks, and to provide minimal control and identification functionality. RTP and RTCP are designed to be independent of the underlying transport and network layers, however it is mainly used with the user datagram protocol (UDP).

D. Java Media Framework

Earlier development of a Java application that use RTP for real-time transmission of media required major programming efforts, however today toolkits and frameworks are available.

The JavaTM Media Framework (JMF, [6] developed by Sun Microsystems, Inc and IBM Corp.) is an application programming interface for incorporating audio, video and other time-based media into Java applications and applets. It makes possible to capture, playback, transmit and transcode audio, video and other media. JMF includes an open media architecture that lets developers access and manipulate various components of the media playback and capture process, or use their own custom plug-in components.

This open architecture enables the introduction of new algorithms into the framework that supports special video compression methods for teleoperation (e.g. defining region of interest – ROI, and use of variable resolution images).

E. Our telerobot system

Our system is a fully functional web telerobot system, which enables to observe and control a special 5 DOF microrobot [7] remotely using a Java capable web browser (Fig. 1). Graphical user interface and live video is provided for the operator to formulate commands and verify the proper execution.

The telerobot is composed of the microrobot and two video cameras connected to the server PC. This server communicates with the client (which can be any computer connected to the Internet, with a Java-capable web browser) by RTP (using the JMF framework for real-time video transmission) and TCP protocol.

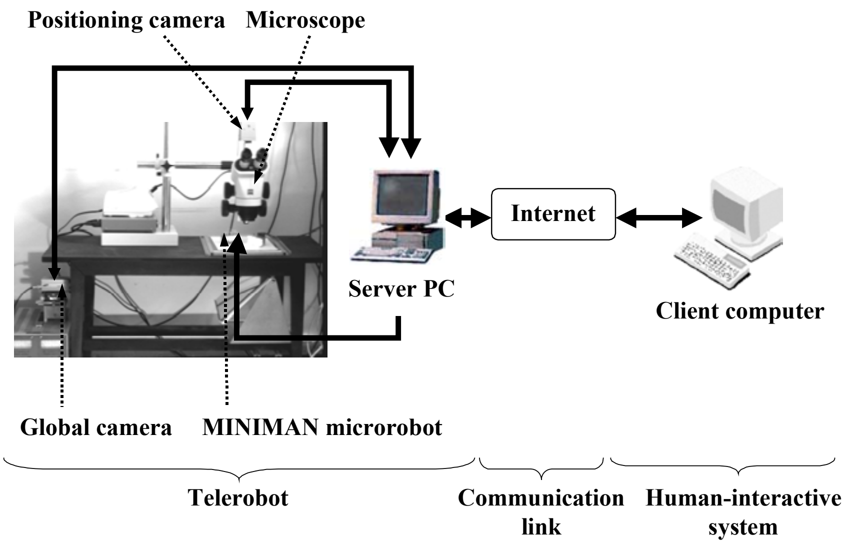

Fig. 1. Architecture of the telerobot system

II. SYSTEM ARCHITECTURE

A. Telerobot

The telerobot contains the following components:

- server PC: with Windows operating system
- microrobot device and its controller board (connected to the server PC through a serial line): it receives move commands from the server and executes them
- positioning camera with a with built-in DSP processor (connected to the server PC through a serial line): monitoring the whole workspace of the microrobot, it mesaures the position of the robot
- close-up camera (connected to the server PC using a BT848-based framegrabber card): this camera is mounted on a microscope and monitors a part of the workspace, this view is displayed for the user

1) Server PC

The controller PC is communicating with the clients via an Internet connection, which makes it possible to observe and control the robot remotely using a Java-supporting web browser. To ensure portability and flexibility, the software of the controller PC is based on the object-oriented, open-source Adaptive Communication Environment framework (ACE, [8]) developed at the Washington University, St. Louis.

2) Microrobot

The microrobot (Fig. 2.) has an aluminum body of 30mm x 30mm x 60mm size, which makes 3-5 μm long "jumps" on its piezoelectric legs by the stick-and-slip method [7]. Its actuator is a small pipette, which is driven by three other piezoelectric legs.

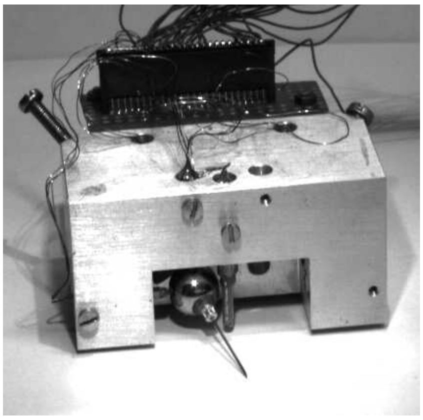

Fig. 2. The MINIMAN microrobot

The controller circuit provides the high-voltage signals that move piezoelectric legs, according to the commands of the controller PC.

3) Positioning camera

The positioning camera detects the markers attached to the robot body (Fig. 2.), measures their position and transmit the results to the controller PC.

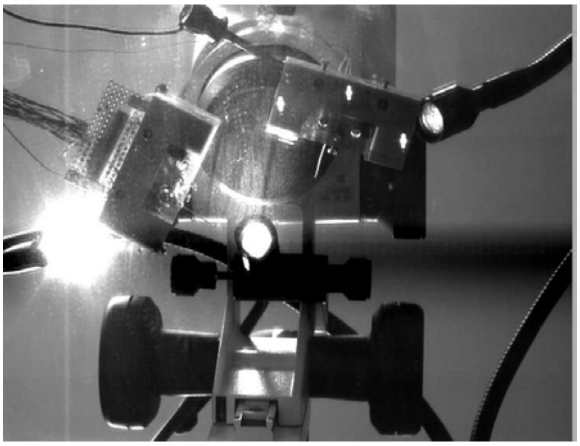

Fig. 3. View of the positioning camera

B. Web client

The web client is Java applet running in a web browser. It implements the user interface of the system, and manages the communication with the controller PC.

The user can move the robot by pointing to a destination position (in this case the robot controller navigates the robot with the help of the positioning camera), or by direct movement commands (giving direction and speed).

The view of the robot's workspace through a microscope (from the close-up camera) is displayed for the user, as well as a schematic draw of the whole workspace (constructed from the measured position values).

C. Communication

Various messages are exchanged between the telerobot and the web client: non real-time administration and configuration data, and real-time commands, position feedback and image data.

1) Transmitted data

These different types of information require different transfer methods, and the optimal solution is also dependent upon the quality of the network segment, the temporary traffic circumstances, the total amount, compression and importance of the specific data.

In the telerobot system the following data have to be transferred:

- administrative data (access control, configuration data): small packet size, requires reliable transmission, no real-time constraints
- commands: small packet size, requires reliable transmission
- control signal (commands, measured position data): small packet size, requires periodic transfer in real-time, packet losses can be acceptable
- video data (visual feedback from the telerobot): requires significant bandwidth, periodic transfer in real-time, packet losses are acceptable

D. *Strategy*

We used stream (TCP) and datagram (RTP over UDP) sockets for communication. Datagram-based communication is based on a non-reliable service, but imposes less overhead than TCP does and transmission delay is more predictable, therefore it is more appropriate for real-time applications.

Therefore video data and control signals are transmitted with RTP protocol (in UDP packets). Video data use the Java Media Framework to transfer the images with JPEG or H263 coding, control signals are transmitted in simple RTP packets without compression.

Commands are transmitted with TCP because reliable transmission is a requirement, and other administrative and not time-sensitive data are also use TCP sockets.

III. Conclusions

We designed and implemented an experimental system to test and evaluate the performance and usability of a modern web telerobot architecture. We used real-time supporting protocols to transmit time-sensitive data and applied compression on high-bandwidth video data.

According to the results of our preliminary experiments with this system it can be concluded that current Internet and web technology enables teleoperation of robotic devices through WWW, though the low quality of Internet services requires the use of special real-time protocols and compression.

IV. References

[1] Sheridan, T.B. (1992) Telerobotics, automation and human supervisory control. Cambridge, MA: MIT Press.

[2] Fiorini, P., Oboe, R., “A Design and Control Environment for Internet-Based Telerobotics”, The International Journal of Robotics Research, Spring, 1998.

[3] Bao, Y., “Transmission of Real-Time Video Across the Internet using MPEG Encoding”, http://www.eecis.udel.edu/~bao, 1997.

[4] Drascic, D., “An Investigation of Monoscopic and Stereoscopic Video for Teleoperation”, Thesis work, Department of Industrial Engineering, University of Toronto, 1991.

[5] Audio-Video Transport Working Group, “RTP: A Transport Protocol for Real-Time Applications”, Request for Comments: 1889, January 1996

[6] Java Media Framework http://java.sun.com/products/java-media/jmf/

[7] Vajda Ferenc, Kleinheincz Gábor: Flexibilis mikrorobot fejlesztése miniatürizált ipari technológiákhoz. TDK-dolgozat, Budapest, 1996

[8] Schmidt, D. C. “The Adaptive Communication Environment, an object-oriented network programming toolkit in C++”, http://www.cs.wustl.edu/~schmidt/ACE.html, 2001.

[9] Fischer, T., Santa, K., Fatikow, S., “Sensor system and powerful computer system for controlling a microrobot-based micromanipulation station”, Journal of Micromechanics and Microengineering 7/1997.