Internet-based Telerobotics: UTM's Experience and Future Direction

Shamsudin H. M. Amin [1]
Rosbi Mamat
Mohamad Fauzi Zakaria
Norhayati A. Majid
Lim Cheng Siong

Center for Artificial Intelligence and Robotics (CAIRO)
Faculty of Electrical Engineering
Universiti Teknologi Malaysia
81310 UTM Skudai
Johor Darul Takzim, Malaysia
Tel: +607 557 6160
Fax: +607 556 6272

[1] sham@suria.fke.utm.my

László Horváth [2]
Jószef Tar

Budapest Polytechnic
H-1081 Budapest Nepszinhaz u. 8
Hungary
Tel. 36 1 333-4513
Fax: 36 1 333-9183

[2] Rudas@Zeus.banki.hu

Abstract

This paper reports on the development of the Universiti Teknologi Malaysia's (UTM) Internet telerobotics systems. The design can be categorised into three phases. In the first phase, the leg of a mobile are tested and controlled through local internet connection. The task is called robot-oriented control system. The second phase of the project is the implemention of the web-based robot using Rhino robot and the last phase is future works on task-oriented of Rhino robot through the Internet.

Keywords: Internet-based telerobotics, robot-oriented, task-oriented.

1. Introduction

The goal of our project in the telerobotics area is to discover and develop the system by combining network technology with capabilities of robots. Using Internet technology for telerobotic application offers the advantage of low-cost deployment. There is no longer a requirement for expensive purpose built equipment at each operator's location. Almost every computer connected to the Internet can be used to control a teleoperable device. The downside is the limitation of varying bandwidth and unpredictable time delays [1]. These Internet features should be defined and considered before designing an efficient telerobotic system. Besides that, several functional requirements should also be defined before designing any telerobotic system:

- Who are the *users* of the specific application and what are their experiences, aptitudes, motivations, and needs?
- What is the *task* for the system and what is required to do it?
- What is the *environment* in which the application will be used, and what is the context in which the task will be done?

Defining the user, task, and environment is essential in specifying appropriate technology for user operator interaction in general and in creating usable systems [2].

In this paper we focus mainly on the overview of our past and recent projects and present some preliminary results.

2. UTM Telerobotics Project

The user interface of telerobot has two ways to be launched in the internet either by using an application or a web interface. Interface using an application usually used a client-server model. The client-server model provides a convenient way to interconnect programs that are distributed efficiently across different locations. Whereas, web is designed as a hyper-text distributed information storage system for technical documentation [3]. Data is stored in many servers, and can be accessed by many clients, seemingly simultaneously. The client programs used are usually referred to as Web browsers because they allow a user to explore inter-related data on different topics.

2.1 Non-web-based Telerobotics System

Our early telerobotics system used an application interface which is based on client-server model [4]. We call this non-web-based telerobotics system. This basic interface is designed using Borland C++ Builder, an object-oriented programming environment which provides a Visual Component Library which is needed to generate the graphical user interface for this interface. Data is transfer using windows sockets element.

System overview

The architecture of the basic system is shown in Fig. 1. The hardware set-up consists of a leg of a mobile robot, which has three joints that move in three degrees of freedom. The leg is controlled by a PC which also acts as a local server. The motor drivers and interfacing electronics are connected to the PC through the parallel port. Every single command that is send to the server interface will invoke the server to perform a certain task to the robot. That is why we call this task as robot-oriented.

The computer is also connected to a main server which has many computers connected to it. Other computers can act as clients and can download the interface and control the motion of the leg of the mobile robot through the local area network. When the system goes to the public network, other users from anywhere can control the robot. Windows sockets provide connections based on the TCP/IP protocol.

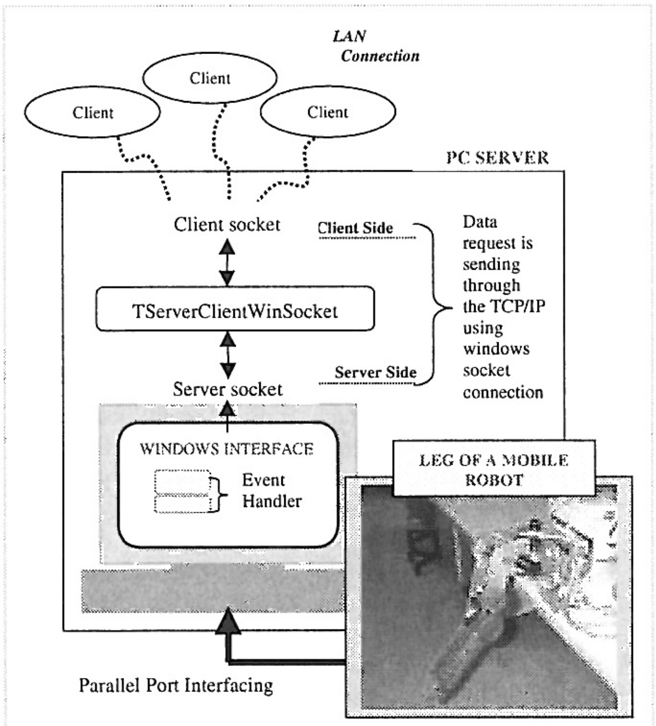

Fig1. : System Architecture for non-web-based

Graphical user interface

One of the most important components of any telerobotics system is the user interface. The display in the user interface should be designed sufficiently so that the user receives enough information about the remote environment. The preliminary basic design of the graphical user interface (GUI) is shown in Fig. 2. This GUI consists of a few panels including motor drive controller panel, speed control panel, selection of either to be a client or a server and indicator panel. Image feedback is shown in Fig. 3.

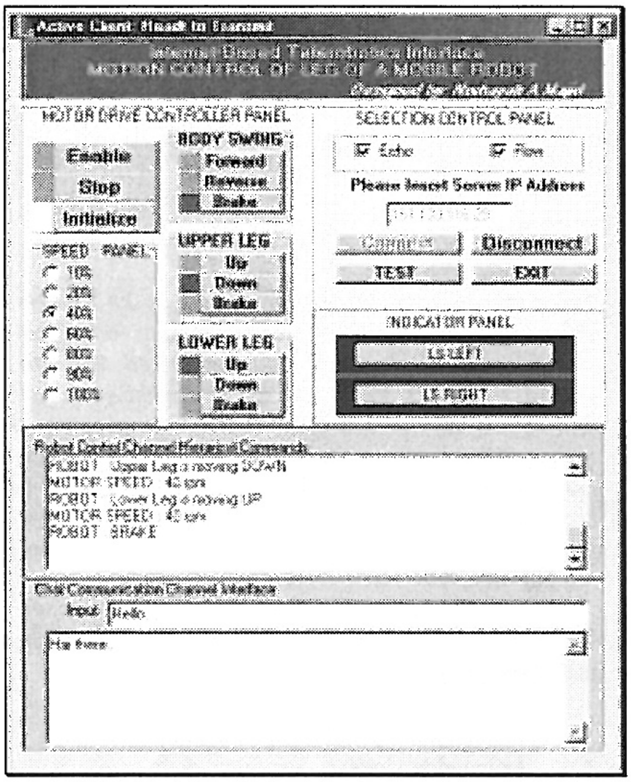

Fig 2: Graphical User Interface for Non-web-based Telerobotics System

Fig. 3: Image Feedback

System achievement

The client can control the robot through the graphical interface and the movement of the robot can be seen through the digital camera that is focused on the robot, which is provided by Microsoft NetMeeting software. In the real situation the client cannot see the robot movement directly because of the delay that always occurs in the Internet application. The system is a robot-oriented system where the client and server will behave like a standalone application. The user must download and execute the application program from the server. The user will be allowed to control the robot after obtaining permission from the server. The interface has been tested locally but is not yet freely available on the Internet yet.

2.2 Web-based Telerobotics System

Our current telerobotics project are based on web interface. A web-based interface is usually a platform independent hypertext mark-up language (HTML) form that is coordinated with a server side common gateway interface (CGI) program [5]. Web browser forms allow the designer to distribute the interface in a platform independent manner with little or no programming application on the interface side. The HTML language contains several different window system components that mimic some standard user interface components. This method is chosen since interfaces can be accessed easily in the web browser and there are multiple platforms in which it can be distributed .

2.2.1 Robots Oriented Interface

Robot-oriented telerobotic is a system that requires the operator to control the robot step by step in implementing a task. In UTM, robot-oriented telerobotics system was built to perform simple tasks such as to move a small plate of steel, which is used in education and entertainment (edutainment) purposes.

System overview

The UTM robot-oriented telerobot system basically consists of three main hardware components that must be integrated. These comsystems are robot system, camera system and host computer system [6]. The robot system includes robot controller and its arm. The robot is a fixed base Rhino XR-4 robot that has five degree of freedom and a gripper. The camera system has two cameras, they are a Sony EVI-D31 pan/tilt/zoom camera type that is used to capture the entire robot environment and a webcam that is attached on the gripper. These systems must be integrated together before being launched to the Internet by programming in the host computer server. System architecture of this telerobot is shown in Fig. 4 and its explanation is given below:

- ***Control server***

The Control Server handles instructions and feedbacks to/from robot controller. The instructions command should be sent to controller via serial communication port if we want to move the arm and know the position of robot.

- ***Image server***

Visual feedback server that controls the camera images feedback before launching Internet server. The actual image of robot depends on type of camera used.

- ***Web server***

The web server system provides three basic services. These three services are login service, system manager service and Common Gateway Interface (CGI) script service. The login service provides communication with the telerobot system by requesting a password and allows the system manager to get information on established connection. This part is important to allow system manager to schedule the user to control the telerobot system by following the database. The CGI script is used to integrate the control and visual feedback server before launching to the client site through GUI.

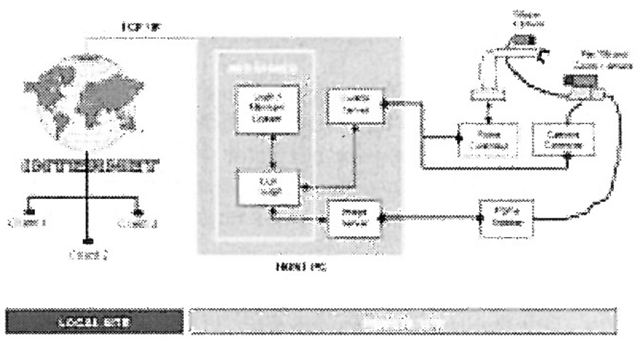

Fig. 4: Web-based Telerobotic System Architecture

Graphical user interface

Robot-oriented GUI shown in Fig. 5 is developed using HTML and C++ Builder. The Robot Control Panel is launched to the web after it is programmed in CGI. HTML was used to integrate CGI – Robot Control and Cameras Live Image page to one web page by using FRAME and IFRAME technologies [7].

System achievement

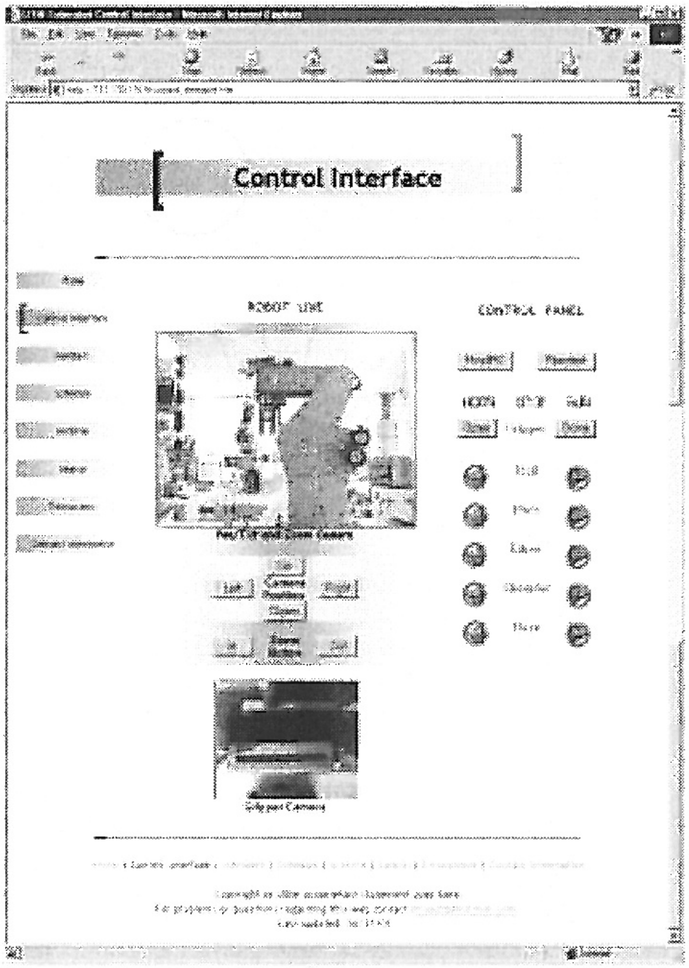

Fig. 5: UTM Telerobot web-based interface

The GUI for this robot-oriented telerobotic system shown in Fig. 5 has been successfully tested on Internet explorer and Netscape navigator web browser. There are some problems especially in internet response time and difficulties to achieve the task target are to be overcome.

2.2.2 Task Oriented Interface

Task-oriented robotic system or so called "task-centric" robotics system requires only the operator to specify the tasks to be done by the system and the system will then plan and carries out a series of action to complete the tasks. In contrast, robot-oriented system will require the operator to plan the actions step by step to get the tasks done. Compared to a robot-oriented system, task-oriented robotic system has higher degree of autonomy. Table 1 shows the comparison between both of the systems.

Advantages of the task-oriented system

The task-oriented internet-based telerobotic system provides better solution to the problems mentioned in previous discussion.

i) *Easy to operate*
Basically task-oriented robotic system is easier to be operated than robot-oriented system since one task in task-oriented robotic system may equal to a set of commands in robot-oriented system. For example the task to move a cube from one location to another which may require the operator to specify a set of commands to move the various motors in robot-oriented system.

ii) *Response time*
Certain processes such as command and task pre-processing will be carried out on the client site thus reduce the waiting time for the response from the server. Furthermore the system may carry out the steps in completing the task without delay between the steps compared with robot-oriented system where each step followed must be specified upon completion of the latest command.

Table 1: Comparison between robot-oriented system and task-oriented robotic system

Robot-oriented System	**Task-oriented Robotic System**
Basic command unit: • Based on robot movement, e.g.: a) Arm type robotic system: shoulder up 30°, elbow down 30°, gripper open or spray start; b) Mobile robot: move forward 30 cm, turn left 45°. • Usually, 1 basic command unit equals to 1 robot instruction.	Basic command unit: • Based on the task designed for the robotic system, e.g.: a) Welding/spray painting system: spot, straight, arc or follows certain marks/pattern; b) Robotic goods sorting system: transfer objects type A to line A and objects type B to line B; c) Mobile robot: find the target such as heat/light source in unknown environment. • Usually, 1 basic command unit equals to a series of robot instruction.

The system can directly convert the command given to robot instruction since 1 basic command unit equals to 1 robot instruction.	The systems need to have the ability to "understand" the task given (requires task specified method) before the task can be converted to a series of robot instruction.
Human will act as path planner to complete the task such as welding and spray painting.	The task controller will do the path planning once "understand" the task(s) required to be done.
Autonomy level: low.	Autonomy level: higher (with certain limitations).
Low efficiency in completing the work since every steps involved must be manually planned/programmed.	Higher efficiency in completing the work since task controller will do the path planning.
Image capturing system (if involved) usually works merely for visual feedback.	Image capturing system (if involved) works not only for visual feedback but also as part of the vision system.
Less complicated to be designed and developed.	Complicated to be designed and developed especially the task controller.
Suitable application: usually for repeated/routine work especially in mass production.	Suitable application: usually for the work that is not/less repeated or the work with uncertainties such as goods sorting where the objects may vary in size, shape, orientation and location.

System architecture

The system is built based on the task-oriented robotic system concept. The task of the system is to manipulate the cubes in front of the robot. The operator only needs to tell the system what to be done (task) rather than how to do it. The operator can tell the system to move some of the blocks to certain locations as well as the pattern of arrangement. Then the system will plan the path on which cube is to be best moved first than the other as well as how the gripper will move the cube.

Fig. 6 shows the system architecture without providing the web service. The preliminary GUIs design is shown in Fig.9. The system can accept task-oriented command from the operators either through mouse operation or natural language. The command will then be processed by the command preprocessor – either interpreter or parser. The purpose of the command preprocessor is to remove the illegal commands such as spelling mistake, syntax error as well as to limit the mouse operation. Information such as the number of objects as well as the location and orientation of respective object are required by the command preprocessor.

Once the system accepts the command from the operator to complete the task, the task will then be passed to the task preprocessor. The task preprocessor will do the simulation if the task could be performed by the task controller. This is very important since the system is designed based on task-oriented approach. Apparently not all tasks can be performed by the task controller due to the limitations in the design and the task-oriented robotic system itself. The complicated task may need to divided into sub-tasks with the assistance from the operator
The task will then be passed to the task controller to do the path planning as well as the transformation to action. The combination of the task controller, the robotic system (the robot controller and the robot) as well as the sensory system (sensors and the sensory sub-system) will form a closed-loop task control sub-system as shown in Fig. 7. In other word, the system will be able to carry out the task autonomously.

At the end of the project, the system must be able to provide the web service. Web-based and non-web-based system will be developed for comparison. The non-web-based system architecture is shown in Fig. 8. The task control sub-system mentioned will be kept on the server. An application program will be developed to provide the command and task pre-processing. The application program will run on the client site. With the pre-processing carried out on the client site, this absolutely will reduce the data transferred and waiting time for the response from the server.

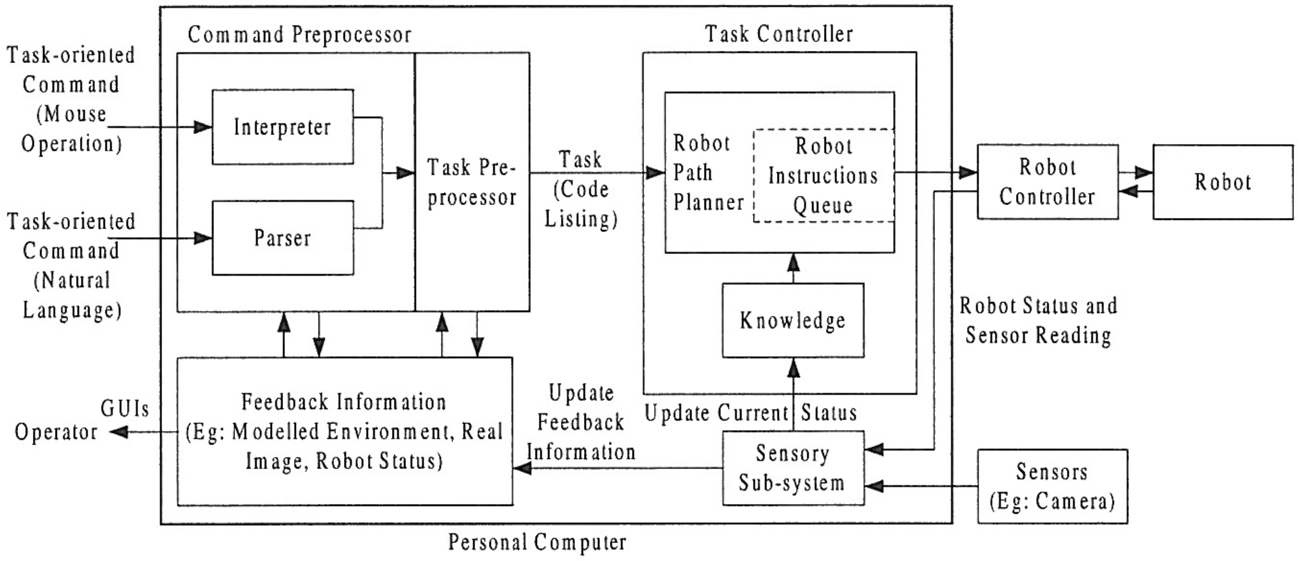

Fig. 6: Task-oriented robotic system architecture (without web service)

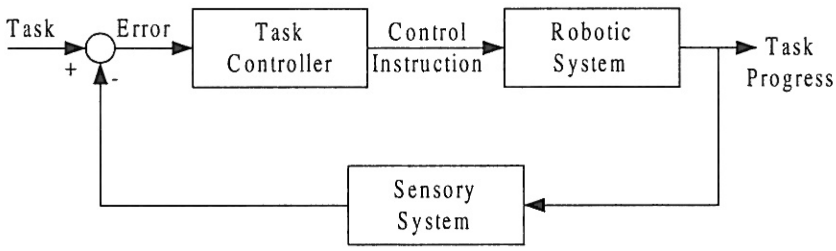

Fig. 7: Block diagram of the task control sub-system (closed-loop)

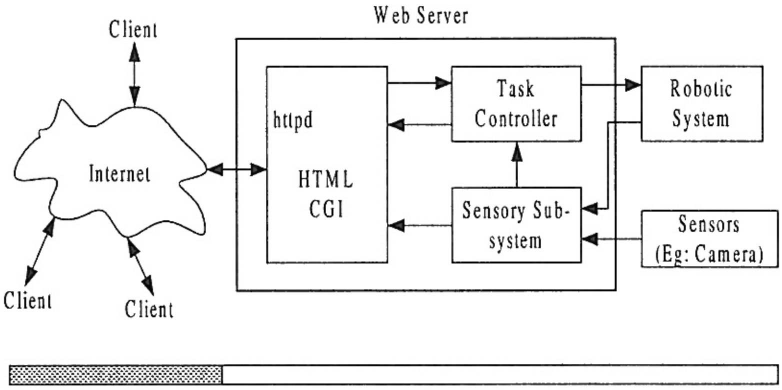

Fig. 8: Internet-based telerobotics system architecture

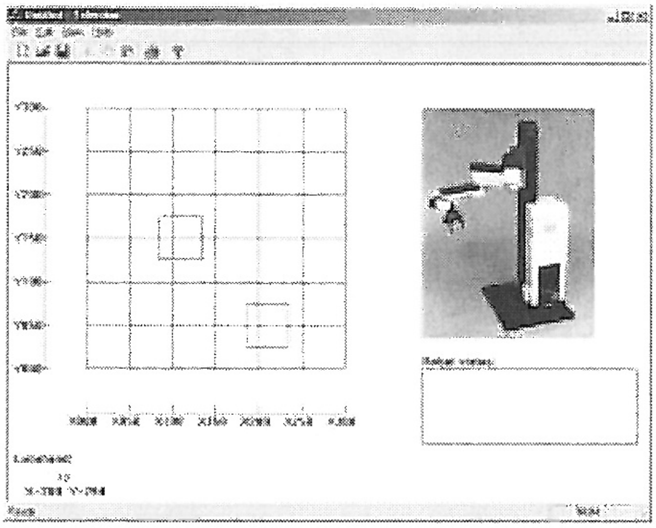

Fig. 9: Preliminary GUIs design (without web service)

3. Future Work

There have been many internet-based telerobotic projects developed since the first robot launched on Internet in 1994. Some of the projects are designed for critical applications such as telesurgery and telemanufacturing. Nevertheless, these applications are too risky and not practical for the current technologies available for the Internet. Unless in the future there are some break through technologies introduced and are low cost and publicly available or the system must be developed based on better quality connection but higher cost such as leased line and fibre optic. In a nutshell, our future direction of internet-based telerobotic projects will tend toward edutainment which is in line with the nature of today's Internet – publicly available, low cost as well as vulnerable and suffered from time delay.

4. Conclusion

We have successfully developed the internet-based telerobotic system for the leg of a mobile robot as well as the fixed type robot. The systems are designed based on robot-oriented and task-oriented concept. The project for fixed robot is expected to be available on the Internet in July. Currently we are developing the internet-based telerobotic system for the mobile robot and the fixed robot based on task-oriented concept. A comparison will be later made between robot-oriented and task-oriented systems.

Acknowledgements

The authors would like to thank the Malaysian Ministry of Science, Technology and Environment for sponsoring this work under IRPA 09-02-06-0022.

References

[1] H. Friz. " Design of an Augmented Reality User Interface for an Internet based Telerobot using Multiple Monoscopic Views", Diploma Thesis, Technical University of Clausthal, Germany, Sept. 1998.

[2] H. Thimbleby. "User Interface Design". ACM Press, New York. 1990.

[3] Norhayati A.M, Shamsudin H.M.Amin, Rosbi Mamat. " Internet Based Leg Motion Control of A Mobile Robot", Proc. of the 2nd International Conference on Advances in Strategic Technologies (ICAST 2000), 15th –17th August 2000, Selangor, Malaysia, Vol.1, pg: 283-292, 2001.

[4] Norhayati A.M, Shamsudin H.M.Amin, Rosbi Mamat. "Development of the Internet Interface for Leg Motion Control of A Mobile Robot", Proc. of the 1st International Conference on Mechatronics (ICOM'01), 12th – 13th February 2001, Kuala Lumpur, Malaysia, Vol.2, pg: 665-675, 2001

[5] Shamsudin H.M. Amin, Rosbi Mamat, Mohamad Fauzi Zakaria, Imre J. Rudas, Laszlo Horvath, Joszef Tar. "User Interface Design for Internet-based Telerobotics". Proc. of SCOReD 2001, CD Vol., 2001.

[6] Mohamad Fauzi Zakaria, Shamsudin H.M. Amin, Rosbi Mamat. "Design and Development of Control System for Internet-based Telerobotics". Proc. of TENCON 2000 Vol. II, pg. 338-342, 2000.

[7] Rick Darnell. " HTML 4 Unleased – Second Edition", Macmillan Computer Publishing, USA, 1998.