STUDENT PROJECTS FOR UNDERGRADUATE MECHATRONICS EDUCATION PROGRAM

Assoc. Prof. Dr. Riko Šafarič, Asist. Prof. Dr. Suzana Uran, Faculty of Electrical Engineering and Computer Sciences, University of Maribor, Slovenia

ABSTRACT

Student projects are an essential part of the Mechatronics education program at the Faculty of Electrical Engineering and Computer Sciences in Maribor. Mechatronics education program and the student projects conducted up to date are briefly described in the paper. Student project with a title Virtual robotic Lab for remote control is described in greater detail. This project enables execution of laboratory assignments on distance via Internet which might give new quality to laboratory assignments for students.

Key words: Student Projects, Learning by doing, Mechatronics education

1. INTRODUCTION

This paper represents student projects for Mechatronics education program at Faculty of Electrical Engineering and Computer Sciences in Maribor. The Mechatronics education program at Faculty of Electrical Engineering and Computer Sciences is new. It was started in the fall of the year 2000/2001. Great effort has been made to make Mechatronics education program attractive for students and to make it comparable to already established Mechatronics education programs abroad [1, 2] when the program was designed.

Investigations have shown that traditional education program is handicapped by the following [3]:

- students are bored and passive during traditional lectures,
- even laboratory experiments are no fun,
- students major motivation is to pass the exams and not to really know the learning material,
- studying is considered as an important part of learning and at last
- incorrect student answers are not desired.

Psychological studies have also shown that people remember only 50% of what they hear and see in comparison to 90% of what they try to realise. On this basis Learning by doing has become an interesting alternative to traditional educational programs and is introduced in educational programs [2] in the form of student projects. The benefits of introduction of student projects into traditional education program is fourfold:

- integration of research components, laboratory development and industrial projects into education is enabled [2],
- learning of practical engineering skills is introduced,
- student team work is introduced and
- interest in engineering is aroused and motivation for learning is increased [2].

But in order to get the desired benefits from student projects these must be well designed. First of all they must be fun for students. Students can choose the project they will work on therefore their motivation for learning is increased due to [3]:

- the control of learning is passed from teachers to students,
- the learning becomes more case oriented and
- more related to their lives and to the real life (team work, practical engineering skills).

The paper is organised as follows. Section 2 introduces Mechatronics education at Faculty of Electrical Engineering and Computer Sciences in Maribor and brief description of student projects up to date. Section 3 represents a student project on virtual robotics laboratory in greater detail. Section 4 draws the conclusion.

2. MECHATRONICS EDUCATION AT FEECS IN MARIBOR

The Mechatronics program at Faculty of Electrical Engineering and Computer Science (FEECS) runs as an four year (8 semester) undergraduate program in co-operation with Faculty of Mechanical engineering, Faculty of Business and Economics and Faculty of Education in Maribor. It started in the fall of the year 2000/2001, as was already mentioned. It is based on a credit system and consists of required and elective courses. Courses of the first two years are the same as for the Electrical Engineering program. Credit system starts in the third year. A student must collect at least 75 credit
points (CP) per year. Courses given in the third and fourth year in the Mechatronics program are shown in Table 1 (see also http://www.ro.feri.uni-mb.si).

Project work in the third year is constituted from a part of the hours of courses on Automatic control, Mechatronic systems, Machine elements and Gears and Drives. Similarly, project work in the fourth year is constituted from a part of the hours of courses on Robotic systems, Communication Technologies, Servo-systems, Intelligent Control in Mechatronics and Mechanisms. Total amount of hours assigned for student project work is approximately 15 % of all hours of Mechatronics program per year. Students could work on the same project which they had chosen in the third year of their study also through the fourth year of their studies and finish their project with their diploma work.

<table>
<thead>
<tr>
<th>Required Courses</th>
<th>Elective Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Management Course (9 CP)</td>
<td>Mechatronic Systems Course (9 CP)</td>
</tr>
<tr>
<td>Automatic Control Course (12 CP)</td>
<td>Logic control Course (7.5 CP)</td>
</tr>
<tr>
<td>Course on Production Technologies (9 CP)</td>
<td>Fundamentals of Robotics (7.5 CP)</td>
</tr>
<tr>
<td>Course on Machine elements (12 CP)</td>
<td>Robotic Course (4.5 CP)</td>
</tr>
<tr>
<td>Course on Gears and Drives (7.5 CP)</td>
<td>Microcontrollers Course (9 CP)</td>
</tr>
<tr>
<td>Robotic Systems Course (7.5 CP)</td>
<td>Course on Internet applications (9 CP)</td>
</tr>
<tr>
<td>CAE/CIM Systems Course (7.5 CP)</td>
<td>Servo-systems Course (9 CP)</td>
</tr>
<tr>
<td>Course on Communication Technologies (6 CP)</td>
<td>Course on Intelligent Control in Mechatronics (9 CP)</td>
</tr>
<tr>
<td>Tele-operating Systems Course (7.5 CP)</td>
<td>Industrial Electronics Course (9 CP)</td>
</tr>
<tr>
<td>Course on Product Development (6 CP)</td>
<td>Power Electronics Course (9 CP)</td>
</tr>
<tr>
<td>foreign language – English or German</td>
<td>Tele-communication Systems (9 CP)</td>
</tr>
<tr>
<td></td>
<td>Real Time Systems (9 CP)</td>
</tr>
<tr>
<td></td>
<td>see more elective courses in [4]</td>
</tr>
</tbody>
</table>

Table 1: Mechatronics education program

Student projects are conducted in the Laboratory for robotics. Student courses with short description which were conducted up to date are shown in Table 2.

<table>
<thead>
<tr>
<th>Student project</th>
<th>Short description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tele-operation via Internet (3 students)</td>
<td>Active Vision System with 2 D.o.F. was built and tele-operation of the system via Internet in order to trace motion of a person or an object was performed</td>
</tr>
<tr>
<td>Mobile Robot with parallel mechanism (3 students)</td>
<td>A mobile robot which climbs up and down the stairs due to its four axis parallel mechanism was built</td>
</tr>
<tr>
<td>Cart and Beam Mechanism (3 students)</td>
<td>A 2 D.o.F. underactuated mechanism consisting of a cart sliding on a beam was built and cart position control implemented</td>
</tr>
<tr>
<td>Virtual Robotics Lab for Remote Control (3 students)</td>
<td>A working facility which enable students to do their laboratory exercises or drive robots in laboratory from home was built</td>
</tr>
<tr>
<td>Bicycle with auxiliary electric drive (3 students)</td>
<td>A bicycle with auxiliary electric drive which could solve traffic/parking problems in the cities was built</td>
</tr>
<tr>
<td>Mobile robot (3 students)</td>
<td>An autonomous mobile robot for exploration of unknown spaces was built</td>
</tr>
<tr>
<td>Planar parallel robot with 2 D.o.F. (3 students)</td>
<td>A prototype of a planar parallel robot with 2 D.o.F. was built and Dynamic analysis performed</td>
</tr>
</tbody>
</table>

Table 2: Student projects

At the end of each school year all the projects are represented to public and the marks are given. At the same time students are encouraged to participate the student contest of the Slovene IEEE conference in Portorož with their project work. In the year 2001 student project Mobile Robot with parallel mechanism won 3. place on the student contest of the Slovene IEEE conference. And last year student project Tele-operation via Internet won 2. place on the student contest of the Slovene IEEE conference.

Integration of Internet or better its the most popular component World Wide Web (Web) takes place at almost every educational institution in the world today and also at Faculty of Electrical Engineering and Computer Science in Maribor. Web is temporary integrated into all courses (also the ones for Mechatronics) at the level where the Web site is
considered to be a collection of related Web pages that are accessible through the site home, according to [5], at the Institute of Robotics of our faculty. In practice this means that students can get for specific Courses from the Web [6]:

- all general information needed (e-mail address, telephone numbers, office numbers, consultation hours of the professor and the assistant, time schedule of laboratory exercises, etc.),
- weekly course outline with links to details on lecture notes,
- literature, detailed lecture notes (linearly written text), home works, laboratory assignments,
- links to several external sites useful to master course topics, etc.

We estimate that present integration level of Internet into our Courses matches current accessibility of Internet among our students. Our laboratories with general student access are almost fully occupied with laboratory exercises therefore they don’t enable good accessibility of Web. On the other side the accessibility of Web from home is still a problem due to extra expenses for the lines or due to Internet long response times during busy hours of a day. But we believe in the future accessibility of Web would dramatically increase therefore it is necessary to develop higher levels of Internet integration into our courses and laboratory experiments. A student project with title Virtual Robotics Lab for Remote Control that helped developing a working facility for off-line programming of actual robot in a remote environment and hands-on training on distance was started in our laboratory. In the project it was specified that as little as possible of additional software to any standard WWW browser should be needed for the remote user [4]. The project is described in detail in section 3. The project was conducted during time period of one school year by three students of Mechatronics, in co-operation with two students during preparation of their diploma work. Students involved in the project met weekly with the mentor of the project to discuss problems and the progress of the project. Through the project we strive to qualify our laboratory for student laboratory assignments on distance which could be given at specific courses. Laboratory assignments on distance might not be independent from the actual laboratory assignments (except in the case of commuting or handicapped students) but may give laboratory assignments new quality. It would enable students to repeat their laboratory assignments over and over again as many times as needed to capture the essence of the assignment without the need of presence of teaching personal.

3. VIRTUAL ROBOTICS LAB FOR REMOTE CONTROL

The virtual laboratory approach is based on the concept that it provides a working facility for off-line programming of actual working robot in a remote environment and hands-on training. It is desirable that the robot simulation should be capable of being executed through any standard WWW browser application, e.g. Netscape Navigator, Microsoft’s Internet Explorer etc [9] and VRML browser, e.g. Cortona or Cosmo Player etc. Standard browsers for the VRML 97 language don't incorporate collision detection between shapes in the virtual world [10]. Because the adopted control strategy does not provide the remote user with immediate feedback from the actual work-cell, it is desirable that some kind of collision detection between the virtual robot and the virtual environment is created to prevent, or to predict, robot collisions in the real world. This problem has been solved by building JAVA oriented collision detection software [11] to assure platform independent approach. It may be solved with use of the complete browser [9] and collision detection software [8] in the C++ language which has capability to decrease the execution time of the complete software so as to increase animation speed as was done in [7] and [8] but it loses the platform independence.

The user must first just connect his computer, via the internet, to the WEB page of the VLAB server. Communication between the virtual robot model of the robotic manipulator, which is viewed by the remote user, and the control system which positions the joints of the actual manipulator is achieved as follows:

- authentication, error checking and user runtime scheduling on the VLAB server,
- the user develops a robot task within the virtual environment on the WEB page of the VLAB server,
- the collision detection routine on users side is performed and if no collision occurs in virtual world, the completed robot task file from the remote user, via the VLAB server, is sent to a laboratory execution computer (a robot workcell),
- execution of the requested task within laboratory workcell, and finally
- collation and return the results via VLAB back to the remote user.

The physical test equipment includes: a WWW network server, a network layer, a robot workcell, and remote user computers. The WWW network server is responsible for processing the requests for information by an external WWW browser, installed on the users remote personal computer, delivering on-line documents and providing access to the robotic and control hardware.

The robot work-cell, shown in Figure 1, allows Point to Point (PTP) motion of the robot. The robot data and environment are constant and are set in the VR software. The motion data is programmed by the user. The work-cell includes the 5 axis educational robotic arm, manufactured in our laboratory which has P-position controllers written by Mathworks’ MATLAB (ver.6.1) / Simulink / Real Time Workshop / xPC Target fast prototyping software. The program is executed by xPC operating software on the personal computer (PC), which has all the necessary software drivers written for power electronic and incremental encoder hardware. The robot controller achieves a constant sampling time for the position control loops, leaving the WWW server free to concentrate on the network interface and services user browser requests, thus overcoming the aperiodic sampling problems.
3.1 Software organisation

Following the success of the earlier experiments [8] and [9], the development of an improved human computer interface, integrating the Java language and VRML language within a non-immersive desktop virtual reality environment, was undertaken in order to help improve the realism and sense of presence the user feels when programming the robot. This simulation tool allows the kinematic and dynamic behaviour of the system to be studied, and permits research into task planning, process synchronisation and the communication issues involved with the control of robotic manipulators. The additional processes which are executed by the VLAB server are shown in Figure 2. Figure 3 illustrates the user's view of the robot model and its associated 'virtual' teach pendant.

3.2 Interface to the server

The remote user is connected to VLAB’s WWW site to register the job using an on-line form. The user details are processed by a CGI program running on the server, to determine user authentication, access control and job queue status.

If the work-cell is available for use then the task is sent to the robot controller, thus allowing the user to view the experiment via an online camera immediately. If the work-cell is currently in use then the users may decide to cancel their job and try again later, otherwise the file is placed in a queue for execution at a later date. In this case an acknowledgement will be returned to the user and the results stored in an on-line archive for retrieval at a later date.

The acquisition of data from within the work-cell takes the form of on-line video footage (at 3-4 frames per second using internet videoconferencing) captured by a digital camera located in the work-cell and numerical data (e.g. positional errors and timing information). Numerical data is collected via a data-logger and is returned at the end of the experiment, which allows the various joint trajectories to be studied graphically off-line.

Real time manipulator control is achieved using a separate computer (PC) control system. However the existing controller requires that set points for limb movement, motor drive characteristics, process status, etc. are provided as a stream of parameters from a host computer, in this case the server, to the robot’s own control system, thus necessitating the development of a replacement command scheduling program written by Mathworks’ MATLAB (ver.6.1) / Simulink / Real Time Workshop / xPC Target fast prototyping software).
3.3 Student Project Results

The user invokes the experiment with simple start in the browser with VLAB WEB address which run on the user side (see Figure 2). The user has to connect the client browser to the VLAB via internet, types the password, choose the right application (there is several other applications: LEGO caterpillar mobile robot, robot arm MA 2000, mobile robot Nomad XR 4000 and our application called School Robot), first. The next step is initialisation which lasts between five and ten seconds. Here, the Java collision detection code and VRML code for the particular application are downloaded to the user computer. After this step, the user sees the teach pendant in his VRML browser and is able to communicate with an “on-line execution part” of the VLAB server. A job scheduling and a data acquisition block of the on-line execution part of the VLAB server, or simply xPC proxy, communicate with the executive PC server, which runs the robot workcell, via internet. The user is able to run the real robot with simple point-to-point (PTP) movement commands and receive the actual position values of the real robot arm. When the new PTP movement command is executed on the user computer the procedure runs as it is described on Figure 4. First, “Java buttons for movement control” are pressed. The command is then checked in the “collision detection” part of the software and if after triangulation, geometric transformation and triangle-triangle intersection test of the robot and environment VRML model no collision is detected in “the virtual world” then command is proceeded via network layer to the VLAB server. If the collision occurs the procedure stops and gives the user an error report.

After the command is executed on the real robot workcell, the actual values of the robot axis positions are sent from the robot workcell via VLAB server to the user computer, where with a help of a VRML browser and a predefined VRML model geometry execution of the command could be seen.

Our approximated School Robot model and its environment (Figure 3) has cca. 400 triangles. The time spent for the execution of collision detection routine was less than 15 ms. The personal computer (Pentium III - 600 MHz, 256 Mbyte RAM, 3D accelerator, Windows 2000) was used as the user computer to perform the experiment. The time needed for VRML model execution was so small that it was neglected in the experiment.

The VLAB server was the personal computer (Pentium III - 1 GHz, 256 Mbyte, Windows 2000) and the executive PC-server was an old fashioned 486 series personal computer. The maximum speed of the Ethernet communication link between both servers was 100 Mbps and also 100 Mbps in the first test (University campus link) and 56 Kbps in the second test (analog modem via public telephone system) between the user’s personal computer and the VLAB server.

Figure 2: The Robot Interface Between a Server and a Client
The time spent to send six desired values from client’s computer to the VLAB server’s xPC proxy and six actual values vice versa was 207 ms for the first test and 514 ms for the second test. Sending six desired values from xPC proxy to xPC control kernel (robot workcell) and receiving six actual values vice versa lasts 317 ms and 163 ms, respectively. The resulting values are the average values (22 measurements, half of them in the rush hours and half of them when internet lines were not busy).

So, the average overall time for the execution of VRML visualisation, collision detection, communication between client, VLAB server and xPC executive server is 702 ms for the first test and 1009 ms for the second test.

4. CONCLUSION

Student projects were introduced into the Mechatronics education program in order to increase students learning motivation, to introduce learning of practical engineering skills, to introduce student team work and to integrate research work, laboratory development and industrial projects in the educational program. A variety of student projects conducted in our laboratory were briefly described and student project with the title Virtual Robotics Lab for remote control was described in detail. Through the Virtual Robotics Lab for remote control we would qualify our laboratory for student laboratory assignments on distance which may give laboratory assignments new quality for commuting and handicapped students as well as for regular students which would be given opportunity to repeat their laboratory assignments over and over again as many times as needed to capture the essence of the assignment.

Figure 4: Execution of VRML robot model and robot model with collision regions