

This article was downloaded by:[Goldstain, O.]
On: 20 December 2007
Access Details: [subscription number 780860632]
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954
Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



European Journal of Engineering Education

Publication details, including instructions for authors and subscription information:
<http://www.informaworld.com/smpp/title~content=t713415994>

Remote learning for the manipulation and control of robotic cells

O. Goldstain^a, I. Ben-Gal^a, Y. Bukchin^a

^a Department of Industrial Engineering, Tel-Aviv University Ramat-Aviv, -aviv, Israel

Online Publication Date: 01 August 2007

To cite this Article: Goldstain, O., Ben-Gal, I. and Bukchin, Y. (2007) 'Remote learning for the manipulation and control of robotic cells', European Journal of Engineering Education, 32:4, 481 - 494

To link to this article: DOI: 10.1080/03043790701337213

URL: <http://dx.doi.org/10.1080/03043790701337213>

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: <http://www.informaworld.com/terms-and-conditions-of-access.pdf>

This article maybe used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

Remote learning for the manipulation and control of robotic cells

O. GOLDSTAIN*, I. BEN-GAL and Y. BUKCHIN

Department of Industrial Engineering, Tel-Aviv University Ramat-Aviv 69978, Tel-Aviv, Israel

(Received 8 October 2006; in final form 11 February 2007)

This work proposes an approach to remote learning of robotic cells based on internet and simulation tools. The proposed approach, which integrates remote-learning and tele-operation into a generic scheme, is designed to enable students and developers to set-up and manipulate a robotic cell remotely. Its implementation is based on a dedicated website that supports a full three-dimensional simulation and a manual online control of the cell. The website provides high flexibility for making changes in the design of the work area, independent of the user's location. The website can be used to further integrate full-scale remote learning labs in collaboration with other organizations.

Keywords: Remote learning; Tele-operation; Simulation; Robotic cells; Internet; Laboratories; CIM; FMS

1. Introduction and purpose

In this work we propose a method and a tool to integrate aspects of remote manipulation of robotic work cells (*tele-operation*) with aspects of *remote learning*. The motivation for such integration is to enable users to practice not only the remote activation of a robotic cell, but also the availability of redesigning and optimizing both the work cell environment and the work plan itself.

Remote control and manipulation of robots has been previously used to perform predetermined tasks, often in a hostile, unsafe, inaccessible or remote environment (Buckhin *et al.* 2002, Siegwart and Saucy 1999). These tasks were usually well predefined in terms of their work elements, requiring the user's focus on an online remote manipulation of the robots rather than on the optimization of the work plan. Accordingly, the interfaces for these tasks were designed to deal with a given environment setting and a set of fixed tasks (Siegwart and Sauch 1999, Chen *et al.* 1999, Ravindra *et al.* 2004, Rohrig and Jochheim 2000, Dougeri and Matiakis 2006, Gil *et al.* 2000, Khan *et al.* 2002).

Remote learning, on the other hand, focuses on the ability to teach the user how to obtain an optimal solution given a flexible set of tasks and tools. Modern technology enables to integrate three-dimensional (3D) simulation tools with a remote control and manipulation interface. This integration can be used, for example, to comply with more complicated tasks in flexible robotic

*Corresponding author. Email: goldstai@post.tau.ac.il

cells (Candelas *et al.* 2005). Remote learning has been an increasingly growing field in the last two decades. The internet development, as well as the increase in personal computer's (PC's) capabilities and bandwidth capacity, has made remote learning through the internet a convenient learning environment that leads to a variety of new learning interfaces and methods (Bukchin *et al.* 2002, Chen *et al.* 1999, Ravindra *et al.* 2004, Bischoff and Rohrig 2002, Calkin *et al.* 1998, Fernandez *et al.* 2005). Integrating simulation and real implementation activities is considered a must in current engineering education (Khan *et al.* 2002, Fernandes and Martins 2001). By enforcing such a paradigm, this paper contributes to the continuous improvement of best practices in engineering education (Calkin *et al.* 1998, Enrique Sucar *et al.* 2005, Michau *et al.* 2001).

The specific problem addressed in this paper is how to enable a number of users (students in our case study) simultaneously to design and optimize the work plan for a remote robotic cell (Fernandez *et al.* 2005). The users are required to learn, through a 'try and error' process, the appropriate way to operate a robotic cell previous to its remote (online) activation. There are several learning schemes that can be used to achieve this requirement. The method we found most efficient, in terms of learning time and efforts as well as in terms of resource usage, is having the users operate a simulation module before the online execution of programs in the cell. According to this scheme, each user simulates the process offline at his free time and then fine-tunes and executes his task online on the work cell. The proposed system approaches remote robotic manipulation from a new angle, enabling users to learn and deal with different robotic cells, as well as experiencing the control and manipulation of such cells from a remote site. The system can be easily adapted to any other field of learning, providing an available simulation program for that field (Henry and Schaedel 2005). The learning process contains simulation and optimization of tasks that require a cognitive learning process, which differs from the required motoric learning process when operating a robotic cell on site (Bukchin *et al.* 2002, Calkin *et al.* 1998, Cooper *et al.* 2002).

Alternative approaches were proposed to carry out the required pedagogical tasks (Bukchin *et al.* 2002, Rohrig and Jochheim 2000, Calkin *et al.* 1998, Tzafestas *et al.* 2006). The first approach that we considered was a *lab-based* approach package. It enables a full remote control of the robot's controller and full access to the simulation programs in the lab computers, while receiving all feedbacks remotely by using designated remote-access software packages (i.e. Virtual VNC). The lab-based approach allows the user to react as if he is physically located in the laboratory, yet, it requires a considerably broad bandwidth from both the client's and the server's computers. It also lacks the possibility of administrating several users simultaneously, and bares with it many security IT problems.

Second to be considered was a *home-based* approach, i.e. having the users to simulate and optimize the entire robotic work plan on their home computer, and only then having them send the final programmed tasks to the laboratory's computers. These programs are executed at the lab computers, while the user can observe the execution through a video feed (Calkin *et al.* 1998). The home-based approach, although safer and lower in terms of bandwidth demand, lacks the interactivity part which is so essential to support a real learning process.

Finally, a third approach was investigated by combining the previous above-mentioned approaches. This approach was named the *web-based* approach – an internet website interface integrated with a particular 3D simulation program to be installed on the user's computer. The developed website is presented below in the case study. It enables us to administrate the different users and control the system usage, while keeping the laboratory's computer invisible to the end users, thus, reducing the safety problems. The integrated simulation tool supports redesign of the layout as well as a 'try and error' process, which are vital for learning and skills improvement. The bandwidth required for this type of solution was found to be significantly

lower than other considered approaches. A survey conducted among 60 users of a beta version of the system reassured these findings by yielding a high level of satisfaction.

The rest of the work is organized as follows. Section 2 describes the modules of the methodology that are required to reach the proposed solution. System design development is addressed in section 3 followed by a detailed case study, including hardware and software considerations in section 4. Sections 5 and 6 discuss respectively the conclusions of the study and further developments for the system.

2. Methodology: basic modules of remote-learning and tele-operation

In this section a generic conceptual methodology is proposed to integrate a remote learning process with tele-operation tasks.

Enabling a student to learn and optimize a work plan in addition to remotely operating given robotic tasks requires more than basic manipulation tools for remote control (Gil *et al.* 2000) – a 3D simulation tool is one of the most popular tools dealing with ‘on-site’ learning (Candelas *et al.* 2005, Enrique Sucar *et al.* 2005). The simulation provides the students with the ability to analyse their performance, and to reorganize and redesign the process up to the required level. The used simulation tools provide another important feature to the learning process, namely, the ability to create and record a program for the cell and then apply it to run the physical system. When using a recorded program, the student can easily implement changes to the system layout or to the sequence of performed actions by using historical knowledge; this knowledge is gathered from previous simulations and system feedbacks, such as collisions, inaccurate point definitions and warnings of reaching work envelope boundaries. Both the simulation and the optimization processes are based on the interaction between the human operator and the manipulation interface of the robot itself. A feedback from the system allows students to change and improve design their program, thus, supporting a learning process and improving the student’s skills and learning abilities.

The basic feedback for a remote operation of a robotic cell is a visual feedback (Eliav *et al.* 2005, Aktan *et al.* 1996, Hu *et al.* 2001). In a local cell setting this visual information is available to the operator directly – simply by looking at the system which is located few inches away. When dealing with remote systems, one requires some visual sensors to provide such information (Bukchin *et al.* 2002); these sensors can be a closed-loop TV system or a streaming video of some kind. The latter provides the operator with somewhat rough knowledge regarding the work cell’s layout and the system state following each phase of the operation. Since It provides the user with a two dimensional (2D) picture of a 3D reality, it is rarely accurate enough to enable fine tuning of the robotic arm, making it hard to achieve an efficient learning process.

While investigating the learning process of robotic cells operation, it was found that the 3D view of the cell is a vital component for the success of the learning process. A common way that was found for teaching robotic operation and programming is through 3D simulation programs enabling the student to learn and experience with a variety of simulated work cells, some of which different from the work cell that exists in the lab. The next step was to find a way to combine the capabilities offered by the simulation program, with basic tele-operation tools, creating not only a remote control interface, but also an actual remote learning mechanism.

In order to exploit the full potential of the simulation tools, detailed data from the work cell is required. Such data can be the actual layout of the cell, coordinates and locations of object in the cell and the online location of the robotic arm during the manipulation of the cell. One way to obtain the missing information, which is suggested here, is activating the simulation

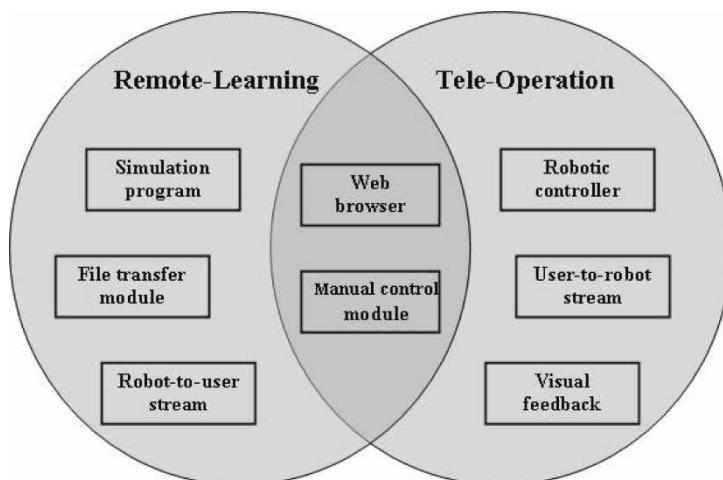


Figure 1. The conceptual integration of remote-learning and tele-operation.

program both at the user's home computer and at the Lab's server that controls the robot. This activation enables a data flow from the robot's encoders to the remote user and provides the user with the necessary information to adjust the simulation and the program it generates.

Figure 1 demonstrates a schematic integration diagram between the remote-learning and the tele-operation modules. Each module contains the main components that are primarily associated with it. Some components that belong to both modules are shown in the intersection area between them. Taking a well-known tele-operation module and combining it both with the data feedback from the controller as well as with the interactivity of the simulation program enables the user to participate in a remote learning process on top of being able to operate a robotic cell. The added value of such integration lies in the capability to apply this design scheme with the necessary adjustments to the available equipment and to different laboratory experiments (Michau *et al.* 2001, Ammari and Ben Hadj Slama 2006). This scheme enables not only the execution of a given lab task, but also a 'trial-and-error' process by multiple simulations prior to execution. We find this scheme to significantly improve the learning phase, yielding a more efficient use of equipment and resources. In the next section we describe and analyse different approaches to implement this integrative scheme.

3. Design alternatives of the system's structure

When considering the design and implementation of a remote-learning and tele-operation laboratory one may consider two alternate approaches. The first one is a '*lab-based*' approach while the other is a '*home-based*' approach. In the following we describe these approaches along with their advantages and disadvantages, while keeping in mind that our objective is to preserve and replicate existing 'on-site' labs for robotic operation learning (Candelas *et al.* 2005, Chiculita and Frangu 2002).

3.1 The lab-based design

The first approach is the '*lab-based*' approach (Tzafestas *et al.* 2006), which is demonstrated in figure 2. This approach uses 'out of the box' programs to gain control of the laboratory

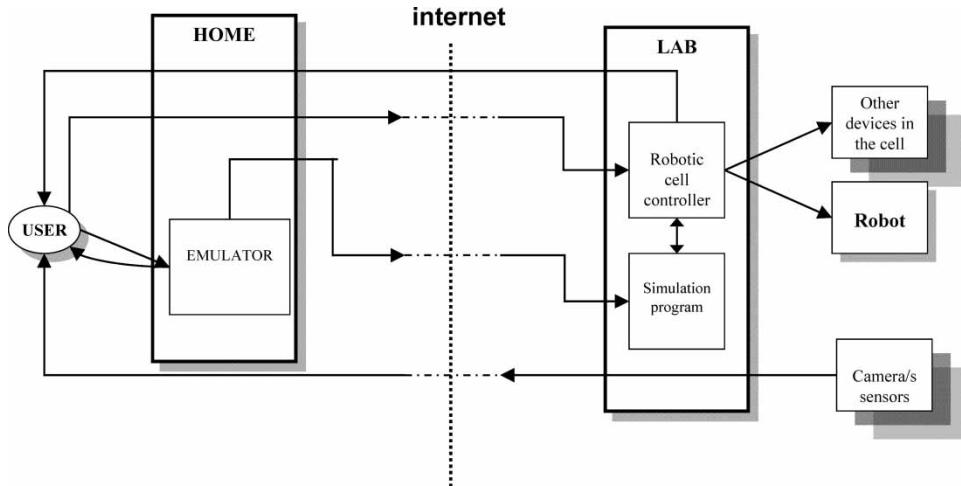


Figure 2. 'Lab-based' design scheme.

computer that controls the work cell. A few programs such as 'windows XP remote desktop control' and 'VirtualVNC' are available for this purpose and known to perform reasonably. These programs emulate the Lab's computer so that the user's keyboard and screen act as connected to the remote computer. The advantage of such an approach lies on its simplicity, providing the user with a quick solution without the need to either download or install any executable programs on their PC. However, working on a remote computer outside the visible area of a work cell requires visual feedback, and adding a camera to the controlled computer can create some technical and methodological problems; the video from the camera is obviously 2D and does not provide the same effect as viewing the cell on site. It also consumes space on the screen, disrupting the display of the simulation program. The bandwidth needed for such approach often cause a bottleneck, resulting in poor response and a low-quality image. Furthermore, this approach does not support a simultaneous connection of multiple users to the system and expose the system to severe IT security threats.

3.2 The home-based design

The second approach is the 'home-based' approach demonstrated in figure 3. This approach provides the user with a simulation program and an accurate scaled 3D model file of a cell's layout, enabling him to initiate the learning at home without being dependent on the robot's availability. In this approach the user sends an already redesigned and tested program to the remote computer, and only then connects to it, as described in the first approach (Calkin *et al.* 1998). This approach lowers the required bandwidth and enables multiple users to operate the simulation tool simultaneously. Still, only one user at a time can gain access to the cell itself. The security threat and the visual feedback problem remain the same as in the first approach, while no real-time data feedback is being gained by this design.

Examining the problems described in the above approaches, one can easily identify the vulnerability of the controller PC as a critical issue, and understand the benefits of keeping the controller PC transparent to the user.

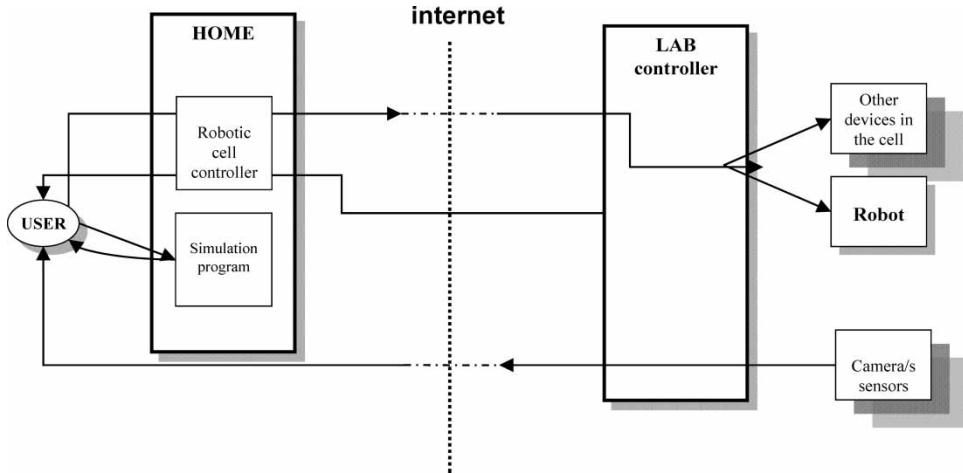


Figure 3. 'Home-based' design scheme.

3.3 The web-based design

A web server that can execute and manage all the communication between the user and the lab is one simple way to achieve the goal of keeping the controller PC transparent to the user (Candelas *et al.* 2005, Yang *et al.* 2004). Installing a server on the lab's computer enables the users to access the lab by communicating through a designed web interface. The visual feedback can also be presented on this website, thus, cutting down bandwidth demands both from the user and the server. The web-based approach, which is shown in figure 4, enables the system administrator to manage the users' connections, and supports better control while reducing associated security threats. This approach integrates the previous approaches according to the proposed methodology in section 2. It includes the following modules:

- a user-client PC running a simulation program and a web browser;
- a web site containing video capabilities, manual movement controls and files-upload protocol;
- a web server running the robotic cell controller as well as the simulation program.

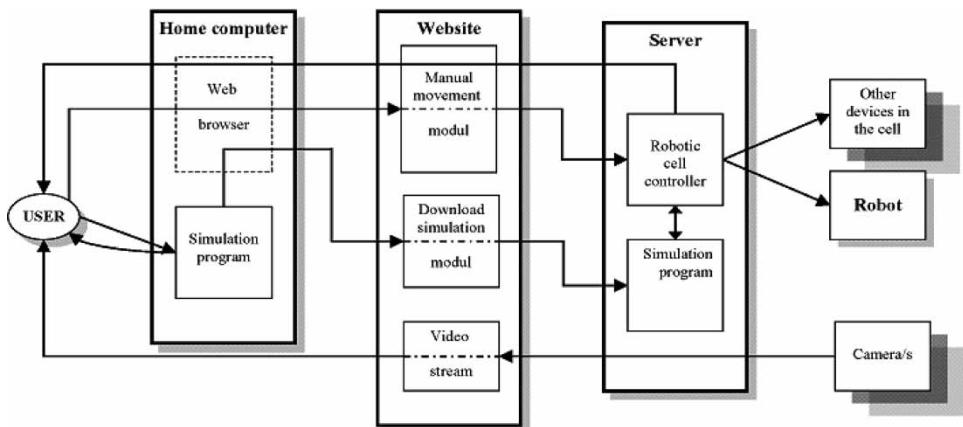


Figure 4. Final ('web-based') design scheme of the proposed system.

Table 1. Comparative evaluation.

Solution type	Simultaneous users	Tools required	Bandwidth	Security	Safety
Lab-based	Single user only	UltraVNC or similar, Web browser	At least 1.5 Mb download (user) At least 1.5 Mb upload (server) for remote connection + extra bandwidth for video server	Full access to all data – poor security	Partially (delayed) online control – good safety
Home-based	Multiple (simulation mode), Single (online mode)	UltraVNC or similar, simulation program, Web browser	At least 1.5 Mb download (user) Bandwidth for video server	Full access to all data – poor security	No online control – poor safety
Website-based	Multiple user administration	Simulation program, Web browser	At least 1.5 Mb download (user) Bandwidth for video server	Controlled access – very good security	Full online control – very good safety

The methodology used in the ‘web-based’ design relies on the interaction between the client and the server (Chen *et al.* 1999, Ravindra *et al.* 2004, Yang *et al.* 2004, Saliah *et al.* 2000). This interactivity was found to be essential for supporting a remote-learning process; it allows a two-way feedback from and to the operator, providing him with the necessary information to improve and refine his actions in order to achieve a more accurate performance of the required task.

Considering the visual feedback module, our main objective is to obtain clear online view of the cell from few different angles. One issue that remains unsolved is the visual feedback gap (3D reality to 2D image) (Doulgeri and Mariakis 2006) that should be further improved. We found that it is almost impossible to perform any kind of task depending only on one video feed that shows the cell from a single angle. Further research is needed to evaluate a multidimensional view, with at least two different angles video feeds, minimize the visual information gap (Yang *et al.* 2004). A human factor study is also recommended to determine the preferred angles and positioning of the cameras.

Table 1 presents a technical comparative evaluation of the above-mentioned approaches. The ‘Tools required’ column shows the main advantage for the ‘lab-based’ solution: no special installation of new software tools is required to implement it. Evaluating the ‘Bandwidth’ column, one can notice that the required bandwidth for the ‘home-based’ solution makes it more attractive than the ‘lab-based’ approach and equally attractive to the ‘web-based’ proposed solution. Considering the rest of the parameters, one can notice the significant advantages for the ‘web-based’ solution: being safer and more secured while supporting simultaneous users. This evaluation supports our decision to develop the proposed system according to the ‘web-based’ approach.

4. Case study

In this section we present a case study of a remote learning website for the robotics laboratory at Tel Aviv University. The website provides a remote controlling interface for the manipulator’s arm. It enables simulating and optimizing the robotic work to be done, recording it, and

subsequently downloading a pre-tested program to the robot's controller. The website enables users to manipulate different objects within the work cell, and to perform numerous different tasks, while receiving both visual and data feedbacks from the lab.

4.1 Hardware and software set-up

The presented system consists of a flexible work cell (SCORBOT, provided by Intelitek Inc.), containing a robotic arm model SCORBOT ER4U (USB controller), a conveyor, a rotating table, three light sensors, an erasable marker, whiteboard and an eraser for the board. The robotic arm as well as the sensors and the peripheral axis (i.e. the conveyer and the rotating storage table) are all connected through a USB controller to a computer, which is also used as a web and video server. The 3D simulated cell on the right-hand side of figure 5 demonstrates the cell set-up of the system. In addition, two internet cameras are connected to this computer.

In order to exploit the possibilities offered by the system, the user has to install a robotic-cell simulation program (ROBOCELL by Intelitek Inc.). Figure 5 shows a screen capture of the program interface that was used in this case study. ROBOCELL is a robotics control software package that provides a user-friendly tool for robot programming and operation. It integrates the SCORBASE robotic control software with interactive 3D solid-modelling simulation software. The virtual robots and devices in the ROBOCELL environment follow the actual dimensions and functions of SCORBOT equipment, providing the users with a fully simulated robotic learning environment and graphical tracking of actual robot operation.

The web interface is programmed using hyper text markup language (html) and hypertext preprocessor (php) programming languages, and is stored on an Apache server on the local computer (Yang *et al.* 2004). Users are administrated with a structured query language (SQL) database, while each user is given a unique username and password. The users are allocated into learning sessions similar to the on-site lab schedule. When a new user is created, a

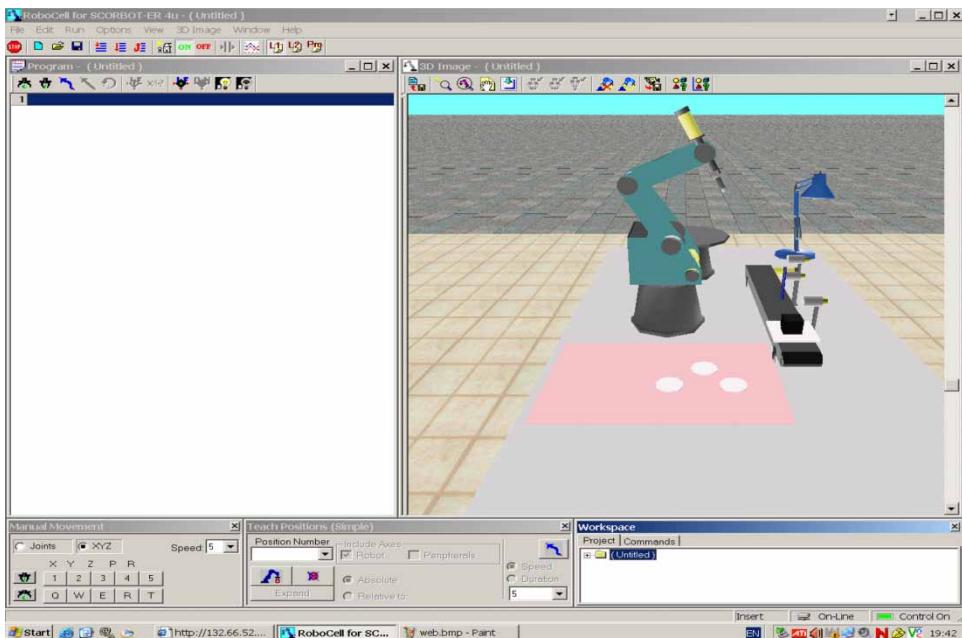


Figure 5. The 3D simulation and programming interface (ROBOCELL for ER4U).

designated folder is being created for them on the server to which their simulation programs and completed assignments are uploaded in a later stage. Users' online sessions are limited by the system to a predefined time frame to avoid blockage of the system. The user can prolong their stay online manually only if other users do not try to access the site. In the latter case, the user is informed of the remaining time frame that is allocated for him until the session will end.

4.2 The system apparatus

The website interface, which is shown in figure 6, consists of three independent features:

- visual feedback, provided by both a streaming video feed and an auto refreshing stills image (right-hand side of the screen);
- a manual movement controller for online manipulation of the robotic arm (top left hand side) – it includes a feedback as to where the end-effector's is exactly located;
- a simulation upload and execution controls (bottom left-hand side).

The work flow of a full learning session for a user is described in figure 7. Before the session starts, the user (student in this case) has to download and install the ROBOCELL 3D simulation program at their home computer, on which the first step of each session is conducted. User manuals for both the ROBOCELL program and the SCORBASE language are available for download at the Lab's main site, including a summarized version for advanced users. In the beginning of each learning session the user is given a task or an assignment to perform. The

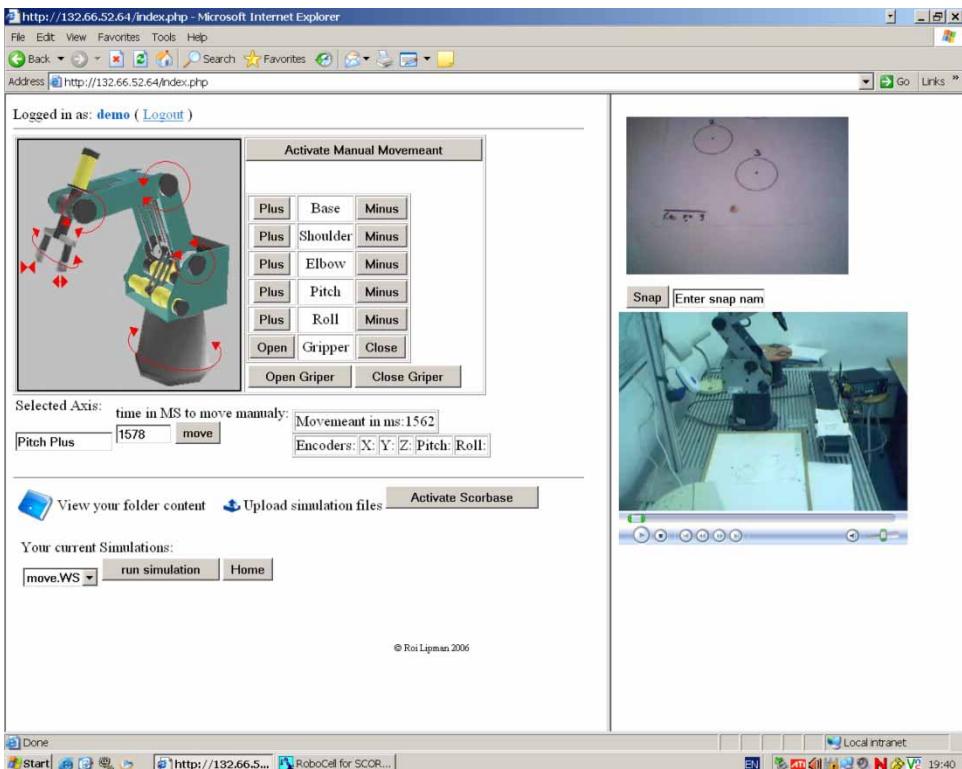


Figure 6. The web interface.

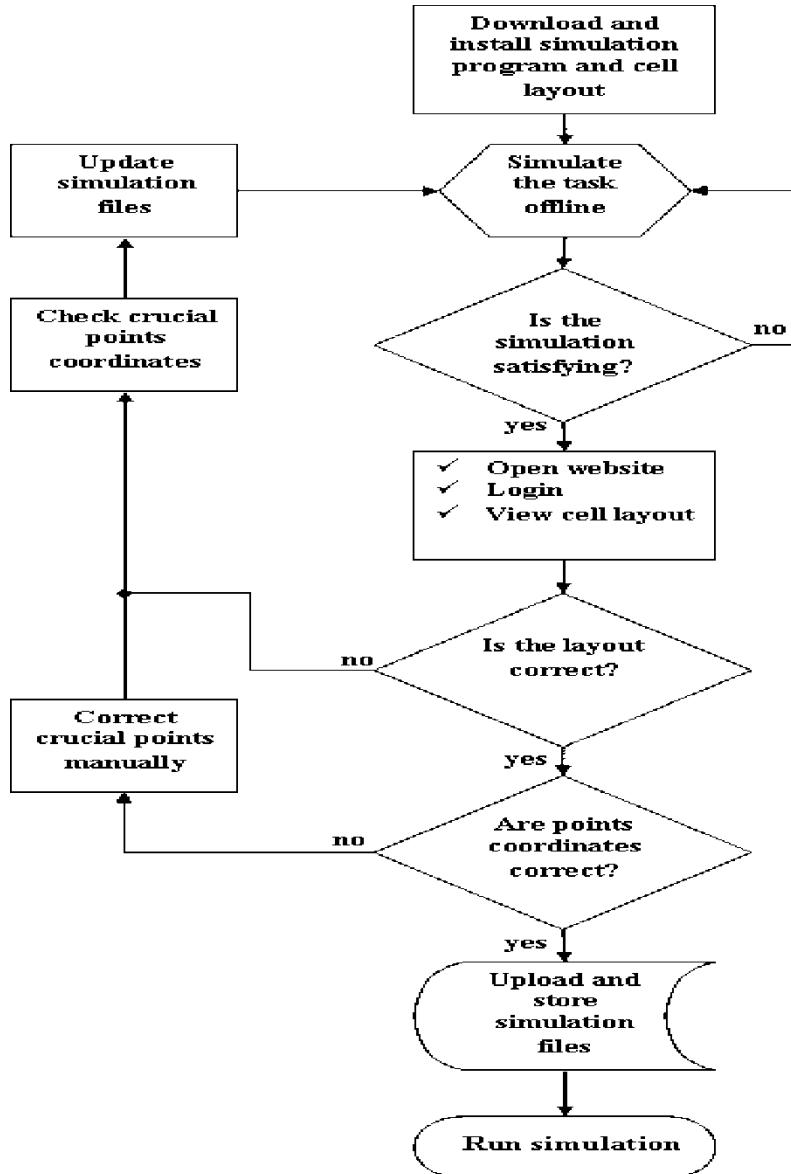


Figure 7. The work flow of an entire remote session.

user is also provided with a cell set-up file of the robotic work cell, on which they will execute his task. This file can be altered in a few minutes, enabling the user to adjust the simulation to changes that occur in the real work cell.

Obtaining all the above modules, the user then starts the 'trial-and-error' optimization stage – a cycle of simulations on the specified layout, which continues until a desired performance of the task is simulated and recorded as a Scorbace program. Once the program has been recorded, the user connects through his web browser to the remote lab learning site. After login, the user enters the web interface page shown in figure 6.

Once logged on, the user is expected to first approve the cell layout – this is done by comparing the simulated layout to the streaming video image. If required, the user has to

make any necessary adjustments to the simulation cell layout. Another required test is to manually move the robot to few reference points in the cell, and to compare the feedback from the encoders to the coordinates in the simulation layout file. The inspection of these reference points is crucial to the calibration phase of the system, as all other point matching during the simulation and execution. This refinement is being executed using the manual movement dialog box available in the interface. The user chooses the relevant axis to be corrected and presses the 'plus' or 'minus' button to increase or decrease the axis's encoder value; the system then measures the time the button was pressed and sends a command to the controller to move the selected axis in the desired direction for the measured time length. If any changes to the simulation file are made, the user must run the simulation again to guarantee that the revised simulation is still valid.

Once the simulation file is optimized, the user downloads four complementary files that were generated by the simulation module to his personal folder on the server. Now, all that remains is pressing the 'home' button on the interface to home the robot, and finally activating the program (by the 'run simulation' button) while following its execution through the camera.

4.3 Users' evaluation

Table 2 summarizes the results of 60 tests performed on the suggested system. The tests were conducted by fourth year students of the Industrial Engineering department at Tel-Aviv University. These students have basic knowledge in robotics programming and internet usage. The data in the table were gathered from questionnaires filed by the students, who experienced both on-site and remote robotic sessions. Four questions were scored on a scale of 1 to 4: where '1' being the lowest evaluation of the question and '4' being the highest (Sivakumar *et al.* 2005).

Four questions were presented to the students. They were asked to give an overall satisfaction score of the system, and score the subjective learning assessment process, the usability and the utility of the simulation module. A total satisfaction score of 79% was obtained for the system.

Table 2. Satisfaction survey summary.

Question	Not at all			Very much score = 4 (%)	Satisfaction score (%)	Average score	S. dev
1 did you benefit from the remote experience	1.7	11.7	31.7	55.0	85.00	3.40	0.79
2 did you improve your skills during the lab	3.3	18.3	50.0	28.3	75.83	3.03	0.79
3 score the remote lab usability	1.7	16.7	55.0	26.7	76.67	3.07	0.72
4 was the simulation helpful	2.4	19.0	40.5	38.1	78.57	3.14	0.81
Total satisfaction score:					79.02		

5. Conclusion

A method to integrate remote-learning and tele-operation is presented in this work. The method is demonstrated and designed in a fully operative case study. This method is applicable on any remote tele-operated environment, depending on the existing 3D simulation tool. The proposed method and interface add a new layer to remote manipulation and control of robots – enabling users to operate in a flexible environment and to perform various changing tasks. Most importantly, using this approach the students participate in an interactive learning process of the activation of these environments without the need to access the lab or the work site.

Further research directions in this area can address the design of the visual feedback, such as how to represent a 3D reality in a web site or a computerized interface. Another issue is how to measure effect of different interface designs on the quality of the learning process. The existing interface is planned to be expanded to support multi-angle visual feedback by several cameras.

A web site for utilizing the remote lab is available at <http://www.eng.tau.ac.il/remote>.

Acknowledgements

This work has been partially supported by VATAT and TAU grant for remote learning. We acknowledge the help of the CIM lab team: first and foremost to Roi Lipman that programmed the website, as well as to Dr Naum Kogan, Eugene Kagan, Israel Sheinberg and Vered Caduri.

References

- Aktan, B., Bohus, C.A., Crowl, L.A. and Shor, M.H., Distance learning applied to control engineering laboratories. *IEEE Trans. Edu.*, 1996, **39**, 320–326.
- Ammari, A.C. and Ben Hadj Slama, J., The development of a remote laboratory for internet-based engineering education. *J. Asynchronous Learning Networks*, 2006, **10**, 12/2006.
- Bischoff, A. and Röhrig, C., Remote experimentation in a collaborative virtual environment. In *Proceedings of the 20th World Conference on Open Learning and Distance Education*, ICDE (International Council for open and Distance Education), Düsseldorf, Germany, April 2001, 2002.
- Bukchin, J., Luquer, R. and Shtub, A. Learning in tele-operations. *IIE Transactions*, 2002, **34**, 245–252.
- Calkin, D. W., Parkin, R. M., Šafaric, R. and Czarnecki, C. A., Visualization, simulation and control of a robotic system using Internet technology. In *Proc. 5th IEEE Int. Advanced Motion Control Workshop*, Coimbra, Portugal, 1998, pp. 339–404.
- Candelas, F. A., Puente, S. T., Torres, F., Segarra, V. and Navarrete, J., Flexible system for simulating and tele-operating robots through the internet. *J. Robotic Systems*, 2005, **22**, 157–166.
- Chen, S. H., Ramakrishnan, V., Hu, S. Y., Zhuang, Y. Ko, C. C. and Chen, B. M. Development of Remote Laboratory Experimentation through Internet. In *Proc. of the 1999 Hong Kong Symposium on Robotics and Control*, IEEE, 1999, Vol. 2, pp. 756–760.
- Chiculita, C. and Frangu, L., A web based remote control laboratory. In *Proc. of 6th Multiconference on Systemic, Cybernetics and Informatics*, Orlando, Florida, 2002.
- Cooper, M., Donnelly, A. and Ferreira, J., Remote controlled experiments for teaching over the internet: a comparison of approaches developed in the PEARL Project. *19th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education*, 2002, pp. 119–128.
- Doulgeri, Z. and Matiakis, T., A web telerobotic system to teach industrial robot path planning and control. *IEEE Trans Edu.*, 2006, **49**, 263–270.
- Eliav, A., Lavie, T., Parmet, Y., Stern, H., Waches, J. and Edan, Y., Kiss human–robot interfaces. *18th International Conference on Production Research (ICPR)*, 2005.
- Enrique Sucar, L., Noguez, J. and Huesca, G., Project oriented learning for basic robotics using virtual laboratories and intelligent tutors. *35th ASEE/IEEE Frontiers in Education Conference*, Indianapolis, IN, October 2005.
- Fernandes, A.S.C. and Martins, M.J., Self-training through the internet. *Eur. J. Engng Edu.*, 2001, **26**, 169–177.
- Fernandez, C., Vicente, M.A. and Jimenez, L.M., Virtual laboratories for control education: a combined methodology. *Int. J. Engng Edu.*, 2005, **21**, 1059–1067.

- Gil, L., Blanco, E. and Aulí, J.M., The virtual laboratory concept applied to strain measurements. *Eur. J. Engng Edu.*, 2000, **25**, 243–251.
- Henry, J. and Schaedel, H.M., International co-operation in control engineering education using online experiments. *Eur. J. Engng Edu.*, 2005, **30**, 265–274.
- Hu, H., Yu, L.P., Tsui, W. and Zhou, Q., Internet-based robotic systems for teleoperation. *Assembly Automation*, 2001, **21**, 143–151.
- Khan, W.A., Al-Doussari, S.M.R. and Al-Kahtani, A.H.M., Establishment of engineering laboratories for undergraduate and postgraduate studies. *Eur. J. Engng Edu.*, 2002, **27**, 425–435.
- Michau, F., Gentil, S. and Barrault, M., Expected benefits of web-based learning for engineering education: examples in control engineering. *Eur. J. Engng Edu.*, 2001, **26**, 151–168.
- Ravindra, T., Huang, L.H., Shi-Jer, L. and Diez, R., Controlling robot through internet using Java. *J. Ind. Technol.*, 2004, **20**, 1–6.
- Röhrig, C. and Jochheim, A., Java-based framework for remote access to laboratory experiments. In *Proc. of the IFAC/IEEE Symposium on Advances in Control Education*, Gold Coast, Australia, 2000.
- Saliah, H., Saad, M., Villardier, L., Assogba, B., Kedowide, C. and Wong, T., Resource management strategies for remote virtual laboratory experimentation. *2000 Frontier in Education Conference*, Kansas City, 2000.
- Siegwart, R. and Saucy, P., Interacting mobile robots on the web. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1999. Detroit, Michigan, USA.
- Sivakumar, S.C., Robertson, W. and Aslam, N. (2005). A web-based remote interactive laboratory for Internetworking education. *IEEE Trans. Edu.*, 2005, **48**.
- Tzafestas, C.S., Palaiologou, N. and Alifragis, M., Virtual and remote robotic laboratory: comparative experimental evaluation. *IEEE Tran. Edu.*, 2006, **49**, 360–369.
- Yang, S.H., Zuo, X. and Yang, L., Controlling an internet-enabled arm robot in an open control laboratory. *Assembly Automat.*, 2004, **24**, 280–288.

About the authors

Ofir Goldstain is an industrial engineer. He holds a BSc (2004) in industrial engineering from Tel-Aviv University and is currently a graduate student in the department of industrial engineering. Ofir has been an instructor in the Computer Integrated Manufacturing (CIM) Laboratory for the last three years. His interests include remote-learning, robotics, vision-systems and human-factors engineering. For extracurricular work, Ofir is the national coach for the Israeli Olympic rifle-shooting team and is a former Olympic shooter in the Israeli national team.

Irad Ben-Gal is a Senior Lecturer at Tel-Aviv University, where he is the Head of the Computer Integrated Manufacturing (CIM) laboratory. He holds a BSc (1992) degree from Tel-Aviv University, MSc (1996) and PhD (1998) degrees from Boston University. He is a member of the Institute for Operations Research and Management Sciences (INFORMS) and the Institute of Industrial Engineers (IIE) and on the Editorial Board of *Trends in Applied Sciences Research*. His papers have been published in *IIE Transactions*, *International Journal of Production Research*, *Technometrics*, *IEEE Transaction*, *Bioinformatics* and other journals. Dr Ben-Gal received several research grants, among them from General Motors, IEEE, the Israeli Ministry of Science and the European Community. He has worked for several years in industrial organizations. His research interests include statistical methods for control and analysis of stochastic processes; applications of information theory to industrial problems; and automation and computer integrated manufacturing systems.

Yossi Bukchin is a faculty member in the Department of Industrial Engineering at Tel Aviv University. He received his BSc, MSc and DSc degrees in Industrial Engineering and management from the Technion Israel Institute of Technology. He is a member of the Institute of Industrial Engineering (IIE) and the College-Industry Council on Material Handling Education (CICMHE) and on the Editorial Board of *IIE Transactions*. Dr Bukchin has held a visiting position in the Grado Department of Industrial & Systems Engineering at Virginia

Tech. His papers have been published in *IIE Transactions*, *Operation Research*, *M& SOM*, *European Journal of Operational Research*, *International Journal of Production Research*, *Annals of the CIRP* and other journals. His main research interests are in the areas of assembly systems design, assembly line balancing, facility design, operational scheduling, multi-objective optimization as well as work station design with respect to cognitive and physical aspects of the human operator.