# Evaluation of Telerobotic Interface Components for Teaching Robot Operation

Ofir H. Goldstain, Irad Ben-Gal, *Member*, *IEEE*, and Yossi Bukchin

**Abstract**—Remote learning has been an increasingly growing field in the last two decades. The Internet development, as well as the increase in PC's capabilities and bandwidth capacity, has made remote learning through the internet a convenient learning preference, leading to a variety of new interfaces and methods. In this work, we consider a remote learning interface, developed in a Computer Integrated Manufacturing (CIM) Laboratory, and evaluate the contribution of different interface components to the overall performance and learning ability of end users. The evaluated components are the control method of the robotic arm and the use of a three-dimensional simulation tool before and during the execution of a robotic task. An experiment is designed and executed, comparing alternative interface designs for remote learning of robotic operation. A teleoperation task was given to 120 engineering students through five semesters. The number of steps required for completing the task, the number of errors during the execution, and the improvement rate during the execution were measured and analyzed. The results provide guidelines for a better design of an interface for remote learning of robotic operation. The main contribution of this paper is in the introduction of a new teaching tool for laboratories and the supplied guidelines for an efficient design of such tools.

**Index Terms**—Telerobotics, simulation, remote learning, interface.

## 1 Introduction

Previous research in the field of interface design for teaching robotics through distant learning is related in most cases to a single application, addressing mainly technical aspects of a specific design. This work is focused on introducing a conceptual methodology supported by technical guidelines for the design of an interface for remote learning of robotic operation. The proposed methodology and guidelines are general and applicable for a wide variety of remote learning tools.

In an earlier work, Goldstain et al. [1] identified different components that are required for a remote-robotic operation in distance learning. The authors developed a new web interface, fully operated in the Computer Integrated Manufacturing (CIM) laboratory at Tel-Aviv University. This interface supports remote learning for control and manipulation of robotic cells.

Having identified the required interface components in [1], the goal of this study is to measure and evaluate the relationships among these components, as well as their effects and usability in the design of a remote learning interface. Such an evaluation is conducted by running a set of laboratory experiments, requiring the users to execute different robotic tasks both locally and remotely, while examining their performance over various interface settings.

The purpose is to enable remote users, via a teleoperation interface, to experience robotic operation as close as possible to actual hands-on operation in the lab.

• *The authors are with the Department of Industrial Engineering, Tel-Aviv University, PO Box 39040, Tel-Aviv 69978, Israel.*
*E-mail: goldstai@post.tau.ac.il, {bengal, bukchin}@eng.tau.ac.il.*

*Manuscript received 6 Mar. 2010; revised 27 July 2010; accepted 19 Jan. 2011; published online 1 Mar. 2011.*
*For information on obtaining reprints of this article, please send e-mail to: lt@computer.org, and reference IEEECS Log Number TLT-2010-03-0022.*
*Digital Object Identifier no. 10.1109/TLT.2011.3.*

The main evaluation tool is a new Test-Oriented-Interface (TOI). The TOI is a web-based interface for remote control and manipulation of a robotic cell. As the study unfolds, elements within this interface are graded and ranked, and their contribution to the remote learning mission is evaluated. The TOI is then mapped into a set of guidelines for designing a remote learning interface. The goal is to maximize the benefits of the interface both for the users (e.g., students), as well as for the institute (e.g., university) providing this tool.

Although several studies on (local and remote) robotic learning are available in the literature, in addition to our description of different interface settings for controlling a robotic arm, this paper evaluates the learning aspects related to the usage of different interface settings, providing an assessment of the various interface components choice.

The remainder of this paper is organized as follows: Section 2 presents a literature review of related distant learning tools for teaching robotics. Sections 3 and 4 describe the system parameters, the performance measures and the design of the conducted experiments. Section 5 presents the results of the experiments, followed by the Discussion and Conclusions in Section 6.

## 2 Literature Review

### 2.1 Remote Learning Interfaces for Robotics

Remote control and manipulation of robots has been previously used to perform predetermined tasks, often in a hostile, unsafe, inaccessible or remote environment [2], [3]. NASA, for example, keeps track of active telerobotic systems providing free access through web browsers; The list of such systems can be found at the NASA telerobotics webpage [4].

An architecture of a WWW-based system for a remote telerobotic operation was presented already in 1999 by Belousov et al. [5]. Their system was mainly oriented for reliability and efficiency, and was based on a 3D Java visualization tool that overcame bandwidth restrictions that existed at the time. In 2001, Belousov et al. [6] presented a similar architecture with an addition of a tool that supports the remote programming of the robot.

Among the many variations of similar systems, one can find Wang and Liu [7] with a teleoperation paradigm for Human-Robot interaction; Hu et al. [8] with a system for the remote control of a robot with visual feedbacks over a simulated map; Kofman et al. [9] with a hand-arm-gesture method for teleoperation; Hu et al. [10] with a pioneering work networked telerobotic systems for teletraining; Ravindra et al. [11] with an interface including Java component and a single camera video feedback for the remote control of robots through the Internet; Siegwart and Saucy [3] with a modular framework for mobile robots on the web, and many other applications.

The tasks in most of the above-mentioned systems were usually well predefined in terms of their work elements, thus, requiring the users to focus mainly on the online remote manipulation of the robots rather than on the optimization of the work plan. Accordingly, the interfaces for these tasks were often designed to deal with given environment settings and sets of fixed tasks [3], [11], [12], [13].

Internet development as well as the increase in PC's capabilities and bandwidth capacity, have made remote learning through the internet a convenient learning environment that lead to a variety of new learning interfaces and methods [1], [14], [15], [16], [17], [18].

Moreover, new integration protocols enable, for example, combining 3D simulation tools with remote control and manipulation interfaces, enabling the management of complicated tasks in flexible robotic cells. Candelas et al. [19] presented a system focused on the training of kinematics and trajectory design for robotic arms. Their work was among the firsts to use a learning platform with full interactivity in the teleoperation process.

Michau et al. [20] presented in detail the expected benefits of web-based learning for engineers. In their work, they express the need for remote learning tools within virtual laboratories, stating that although simulations cannot replace real experiments, remote laboratories provide new ways for practicing hands-on-experiments.

Integrating simulations with real implementation activities is considered a necessity in nowadays engineering education [21], [22]. An example for such an integration is found in Calkin et al. [16] with visualization, simulation, and control of a robotic system over internet technology. This virtual learning mechanism is later referred to by Goldstain et al. [1] as the "Home-Based" design scheme. Similarly, Puente et al. [23], used simulations as a learning tool when suggesting a general system architecture.

In 2004, Yang et al. [24] introduced an internet-based teleoperation scheme of a robot manipulator for educational purpose. Their system integrates a virtual offline simulation with an actual teleoperation module including a visual feedback. In their conclusion they suggest the development of a more general control system, that later was presented in Goldstain et al. [1] as the "web-based" design scheme.

Enrique Sucar et al. [25] refer to virtual tools as a primary step in teaching of robotics. In their work, a virtual laboratory based on simulation was developed and assessed for its usability, yet, without evaluating the required interface design. A modern, fully developed interface for remote learning and programming of a robot arm was also presented by Marin et al. [26].

Reviewing the suggested remote learning systems for telerobotics, the required components in such systems are now reviewed.

## 2.2 Remote Interfaces Design

Siegwart and Saucy [3] describe the basic interface specifications, and address the major difficulties when designing an interacting platform for a remote environment. Their suggested modules include a video feedback module, a robot guidance module, and a virtual representation module. These modules appear, partially or as a whole, in later designs of virtual and remote interfaces throughout the literature [27]. Khan et al. [21] show that both virtual laboratories and remote learning experiments help to ease down the dynamics of laboratory operations.

Enabling a user to learn and optimize a work plan in addition to remotely operating given robotic tasks requires more than basic manipulation tools for remote control [21]. A three-dimensional (3D) simulation tool is one of the most popular tools when dealing with "onsite" learning [19], [25]. Candelas et al. [19], Ravindra et al. [11], Tzafestas et al. [28], Marin et al. [26], and others offered different variations of both offline and online 3D simulation tools. More advanced simulation tools, like the one used in Goldstain et al. [1], provide another important feature for the learning process, which is the ability to create and record a program for the simulated system and then apply it to run the physical system itself.

The basic feedback for a remote operation of a robotic cell is a visual feedback [10], [29], [30]. In local cell settings, such visual information is available to the operator directly simply by looking at the system which is located a few inches away. When dealing with remote systems, one often requires some visual sensors to provide such information [2], [31]. Using the virtual laboratory concept, this feedback is gained through a 3D model, as implemented in Belousov et al. [5] and Belousov et al. [6] using Java.

Tzafestas et al. [28] compare virtual (offline) versus remote teleoperated learning tools. In their work, the requirement for remote learning is a visual feedback in terms of a closed-loop TV system or some kind of a streaming video. Since such visual feedback provides the user with a two dimensional picture of a three dimensional reality, it is rarely accurate enough to enable fine-tuning of the robotic arm, causing difficulty in achieving an efficient learning process. Another more advanced feedback for robotics, is presented in Goldstain et al. [1] and Tzafestas et al. [28]. This is the positional feedback, providing the user with valuable information regarding the positioning of each robotic arm axis. This type of feedback can be used to reconstruct the robot's movement, or even to completely reevaluate the robotic cell layout.

In 2002, Adams [32] suggested critical considerations for human-robot interface development. His concepts of user-centered design and situation awareness guided us in the design of the proposed framework.

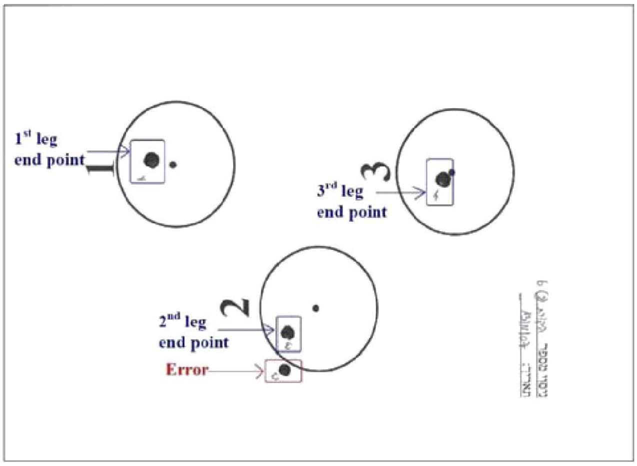


Fig. 1. An example of a completed worksheet.

Goldstain et al. [1] presented a methodology, which is used for the design of the experiments in the next section. The suggested methodology is based on the framework presented by Yang et al. [13] and by Chen et al. [10], preaching for the use of virtual laboratories as an essential tool for learning, prior to the execution of "on hand" experiments.

This work follows the guidelines and suggestions in the above-mentioned papers. It mainly focuses on the analysis of the main components in a telerobotics remote-learning tool. We believe that such an analysis can provide guidelines for an efficient design of such tools.

## 3 The Experiment

The goal of the experiment is to measure and evaluate the effect of different interface components on the usability of the remote learning interface. The next section present the main design factors, the used interfaces, and based on these, the experimental design itself.

### 3.1 The Experimental Task

The designed task for the experiment was a simple "Pick-and-Place" task. It was adjusted to suit a remote operated system in the following way: the users were instructed to manipulate a robotic arm with a marker attached to its gripper in a way that they will reach and mark a dot within three preplaced circles (see Fig. 1). As the learning goal of the task is an efficient operation of the robot while using a remote interface, the users were instructed to try and perform the task in an efficient way as possible, i.e., in the least number of movements possible, and with as little as possible markings outside the designated circles.

The circles were placed at the exact same locations in all different variations of the experiment. The starting point was defined as the "homing point" of the robotic arm.

Three performance measures were considered in the experiments, as discussed next.

#### *3.1.1 Number of Movements Required to Complete a Leg*

In the experiment, a user step (movement) was defined as a single press on one of the controller's buttons. In the examined designs, the movement was recorded as long as the button was pressed, and was stopped when the button was released. Obviously, several movements were required in order to move the robot's arm from point to point. Every single movement was recorded on a data-recording sheet and was associated with one of the legs. A leg was defined as the period between reaching and marking each of the circles, thus, the first leg includes all movements recorded from the homing point until marking the first circle; the second leg was defined as the period between marking the first circle until marking the second circle, and so on.

This steps measure allowed us to assess the overall performance of an operator, and provided us with quantitative inputs regarding the improvement in his performance during the whole task execution.

#### *3.1.2 Number of Errors Recorded*

Errors were defined as a marking made outside of the designated circles (see, for example, the indicated error in Fig. 3). Every mark on the worksheet was numbered and recorded, and the number of markings outside the circles was later analyzed.

The number of errors provided information regarding the complexity of the task or the settings, and helped to evaluate the performance of various interface designs.

#### *3.1.3 Improvement Measure (Learning Curve)*

An improvement measure was evaluated by using the number of movements that were measured in the legs. The learning rate was defined as follows:

$$LR = (N_{total} - N_{H_{to}1})/N_{total},$$

where $N_{total}$ is the total number of movements and $N_{H_{to}1}$ is the number of movements in the first leg (from the homing point to circle #1). Dividing the number of movements required for legs two and three by the number of total required movements was used as a criterion to assess the improvement and the learning pace during the execution. The lower this ratio was, the larger part of the total movements was required by the first leg. In such a case, the subsequent two legs required significantly less movements and, therefore, one could conclude that the improvement from the first leg to the next ones was greater.

### 3.2 Evaluated Design Factors

Three components were defined as main design factors and evaluated through a set of experiments. The design factors were: the use of simulation prior to the task's execution; the use of a virtual real-time presentation (abbreviated henceforth as VRTP) during the task's execution, and the type of control method for the robotic arm. Each of these factors is presented next.

#### *3.2.1 Preliminary Simulation*

The 3D simulation tool, which is used in our experiment, is based on the Robocell software cell-setup module. This tool allows the operator to experiment and learn in advance (i.e., in an offline mode) the robotic system and its environment. It provides the operator with a virtual working cell, similar to the one in the actual site, viewable from all directions and angles. This tool integrates the SCORBASE robotic control software with interactive three-dimensional solid modeling simulation software, replicating the actual dimensions and functions of the real equipment, providing users with a fully simulated robotic learning environment and a graphic tracking view of the actual robot's operations.

The preliminary simulation tool was available in all the considered interfaces and for all the phases of the experiment. It was expected to improve the execution of the task and to shorten the learning period required during the online operation of the robotic cell. For analysis purpose, the different interface settings were examined both with and without the use of preliminary simulation.

#### 3.2.2 Virtual Real-Time Presentation (VRTP)

In certain settings, the 3D simulation tool can be operated not only in advance but also during the online execution of a task. When operated in an online mode, we refer to it as VRTP. The VRTP tool provides the operator with an extra view of the working area, including the possibility to change the viewpoint in direction and orientation during the actual execution.

While this tool might be considered as unnecessary when operating the robotic cell locally within eye contact, it can provide valuable information for a remote operator, unavailable from the static cameras in use. Unfortunately, due to its technical complexity, this tool is not integrated in most of the currently used remote learning interfaces. For this reason, we did not integrate it into the TOI, and it was not used it as an independent factor available for all the design settings. Instead, for analysis purpose of the contribution of this tool, we considered two different remote designs one with and one without the VRTP tool, and called it, respectively, the VRC and RRC interfaces. When used, the VRC was expected to result in a better learning capability of the user, thus leading to a smaller number of steps required to complete a task, and to a fewer number of measured errors.

#### 3.2.3 Control Method of the Robotic Arm

Two common methods are available for controlling a robotic arm: the Axis-XYZ control method and the Joints control method.

The Axis-XYZ control method allows the operator to move the robotic arm along the axis of an imaginary Cartesian workspace. The linear movement is intuitive, but it requires greater computing resources as the robot's controller needs to calculate the exact direction and force for each joint motor and operate multiple motors simultaneously to achieve a linear movement. For these reasons it was technically impossible to implement the required matrix into the TOI, and therefore the Axis-XYZ control method was tested only in the LRC, RRC, and VRC interfaces that are described below.

The Joints control method, although not as intuitive, is technically and mechanically much simpler. This control method is based on activating a single joint motor at a time, resulting in a nonlinear movement (in the case of a polaric joint). It is more complicated for the inexperienced operator to control a robot using this method, and therefore it was expected to result in larger number of steps required to complete a task. Yet, it was found, as explained below, that the greater attention required for joint controlling a robot sometime results in better learning of the remote executions.

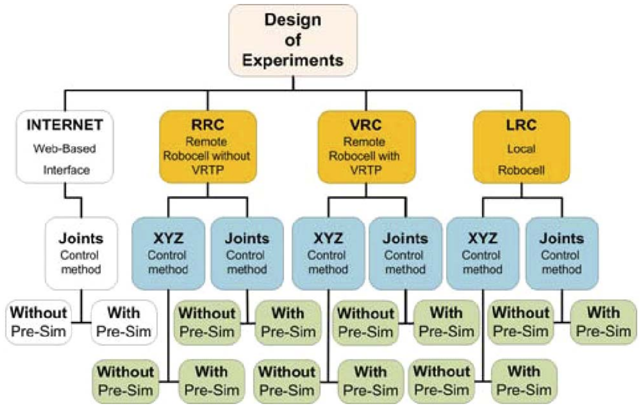

Fig. 2. Phase #1 experimental tree. Comparing three interface types, excluding the Internet interface.

### 3.3 The Compared Interfaces

Two different design platforms were used throughout the experiment: an INTERNET (web-based) platform based on our TOI, as detailed later in Chapter 4.2, and a (wired) Robocell platform, which was operated either remotely or locally for comparison purpose. More specifically, we tested four different interfaces (variations of components) for the evaluation of teleoperation tasks:

1. INTERNET—A Test-Oriented-Interface operated remotely (based on Goldstain et al. [1]).
2. LRC—A Robocell interface operated locally.
3. VRC—A Robocell interface operated remotely, with VRTP.
4. RRC—A Robocell interface operated remotely, without VRTP.

It is important to note that although the LRC is included within the compared interfaces, it is not a remote interface setting. This setting represents the everyday hands-on execution of robot operation locally in the laboratory, and is used here as a benchmark to compare to the other settings, enabling to evaluate their proximity to hands-on operation performance.

The compared interfaces are described in Chapter 4.1.

### 3.4 The Design of Experiments

As not all of the factors were technically possible to be integrated into all the emulated interfaces, a partial factorial design was generated to include the available combinations of factors that could be tested. In order to avoid partial designs, we divided our evaluation tests into two congruent phases, each phase was evaluated as a full factorial model on its own, and together they cover all the design variations that were technically available.

#### 3.4.1 Phase #1 Evaluation Tree

Fig. 2 describes the experimental tree for design phase #1. The transparent (uncolored) branch, representing the internet interface, is excluded from the experiment in this phase since it lacks the Axis-XYZ control parameter.

In this evaluation phase, we examine 1) the effect of the control method on the execution and the learning process

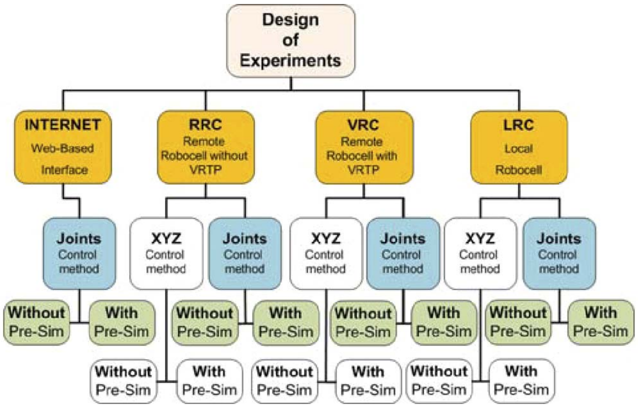


Fig. 3. Phase #2 experimental tree. Comparing four interface types, excluding axis-XYZ Control method.

and 2) the contribution of the VRTP module (integrated in the VRC interface).

### 3.4.2 Phase #2 Evaluation Tree

Fig. 3 describes the experimental tree for design phase #2. The transparent (uncolored) branches, representing the Axis-XYZ control parameter, are excluded from the experiment in this phase to maintain a balanced hierarchical experiment that includes the internet interface.

In this evaluation phase, we examine 1) the difference between the different RoboCell interfaces (LRC, RRC, and VRC) and the designed remote Internet interface, 2) the contribution of the preliminary simulation tool in all the remote interfaces (Internet, RRC, VRC).

## 4 Experimental Settings and Apparatus

Using the "web-based" design scheme described in Goldstain et al. [1], a website interface was designed [33]. This website combines a remote controlling component for the manipulator's arm along with a possibility for simulating and optimizing the work to be done, recording it, and then downloading a pretested program to the actual robot's controller. This website represents an INTERNET interface in the experiments. The Robocell program was used to emulate the other evaluated interfaces as indicated above.

Next, we present the physical layout of the experiment, the design of the TOI and the subjects group used for executing the experiments.

### 4.1 Physical Layout

A dedicated remote workstation was assembled to support the experiments. The workstation was equipped with two screens to standardize the visual feedback size and its position for all the experimented settings.

The first screen was used for visual feedback only, displaying two live video feeds of the robotic cell: an overall view, pictured by a regular camera that captured an isometric view of the cell, and a zoom-in top-view of the work area to follow the accurate operation.

The second screen was used to support the actual teleoperation of the robotic cell. Both Robocell software and an Internet browser were installed in the work station and alternately operated, depending on the experimental stage.

Another workstation was used for the local site and ran the servers of the website and the live video feeds.

As mentioned above, four interface versions were examined, dictating four slightly different settings for the local and the remote stations, as explained next.

#### 4.1.1 Remote Robocell (RRC) Interface

The Remote Robocell workstation was equipped with a dedicated software package that was installed on the remote teleoperation computer and supported the preliminary simulation module. The teleoperation computer itself was physically connected to the robot's controller by a long amplified USB cable. The remote workstation was placed in a remote room preventing eye contact between the operator and the robotic cell. These interface settings were designed for the examination of a modern teleoperation system that can support both the Axis-XYZ control and the joints control methods.

When using the RRC settings during the experiment, the operator got a presentation of the manual movement module of the Robocell software on one screen, and two live video feeds of visual feedbacks on the other screen. In experiments where the preliminary simulation was used, the software was initially set to the offline working mode. This means that the simulation module was presented to the operator for an unlimited time according to his choice before turning into the online operation itself. Once changing to the actual online operation, the simulation module was shut down automatically.

#### 4.1.2 VRTP-Enabled Remote Robocell (VRC) Interface

The VRTP-enabled remote Robocell mainly refers to the online option for simulation with the Robocell software.

The settings of both local and remote workstations were identical to those described for the RRC. The main difference between the two was the ability to keep the simulation module running also during the online operation stage. Such an option enabled the operator to have, on top of the visual feedback of the live feeds, a virtual visual feedback from the simulation module that was updated simultaneously with the movements of the robotic arm.

Potentially, such an option provided the operator with both an advantage and disadvantage with respect to the RRC. On one hand, he could change the orientation, angles, and zooming of the view in the VRTP module, and by that gaining better information than that obtained from the video feeds alone. However, on the other hand, the virtual simulation could never be as accurate as the real video screen, and could have resulted in operation errors, especially when the operator reached the edge of the robot's working envelope.

#### 4.1.3 Local Robocell (LRC) Interface

The local Robocell was used as a control group in the experiments. The LRC ran physically next to the robotic cell itself. In these experiments, the Robocell software was executed without adding any visual feedback. Since the experiments took place within eye-contact distance of the cell, it provided the operator with the opportunity to actually observe the robot and decide on the next required step.

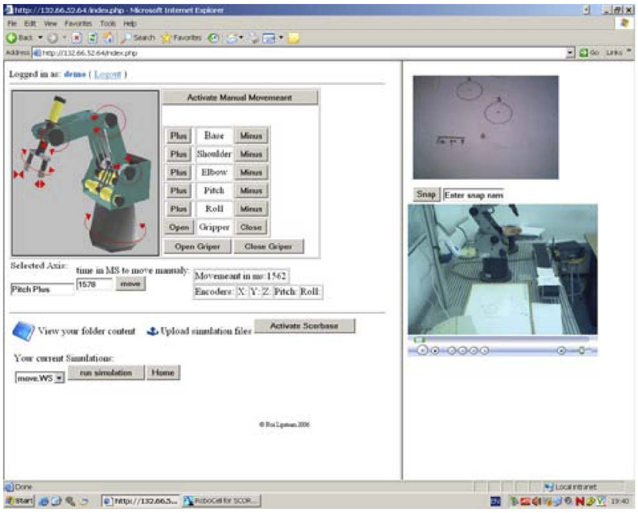


Fig. 4. The TOI interface. The website design consists of three modules combined to give a complete implementation of the proposed methodology.

#### 4.1.4 INTERNET Interface

The Internet workstation contained, in addition to the standardized visual feedback mentioned above, a web browser. The internet browser alone was used to execute the teleoperation at the remote site.

The web browser was used to log on into the proposed TOI and then to remotely operate the robotic arm through it. The visual-feed module in the TOI was not operated during the experiments in order to keep the same visual feedback for all interfaces, as required for analysis purpose. Instead, the operator was provided with a separate visual feedback on the second screen, as happened in all the other remote settings. If a preliminary simulation had to be used in the experiment, the Robocell software was initially operated on the same computer, but only in an offline working mode, and the simulation module was presented to the operator for an unlimited time before turning into the online operation through the website. Once changing to the website TOI online operation, the Robocell software was shut down.

### 4.2 The Test-Oriented-Interface

The web-based interface, shown in Fig. 4, was programmed using html and PhP programming languages, and was stored on an Apache server on the local computer [13]. The designed interface utilizes the Robocell simulation software as a platform [1].

Users were administrated with an SQL database and the sessions were limited by the system to avoid blockage of the system. The code for a single movement was based on measuring the time a specific push button was pressed in the control module, identifying the joint and the direction represented by this button and then sending an operation command through the server and to the robot's controller. The response from the controller, composed of the new encoder values, was presented on the website interface in response. A detailed description of the TOI modules is given next.

Following is the description of these modules.

#### 4.2.1 The Joint Control and Data Feedback Module

The Joint control and the data feedback module refer to the upper left hand side of the interface, shown in Fig. 4. In an endeavor to support the use of the Joints control method, we introduced a Java application in the Internet interface (seen in the gray area of the upper left corner of Fig. 4). This application provides the user with a sequential numbering of the robot's joints and the direction of each joint. A conventional joints controller was also implemented in the interface (middle upper part of Fig. 4) for advanced and experienced users.

There is a major difference between these two controllers and the Robocell interface controller described in the previous section. When using the Robocell controller, the action of clicking on a direction of movement results in an immediate response from the robotic arm, and the robot keeps moving in the desired direction until the push button is released. On the Internet interface presented here, pushing a controller button starts a timer (shown in Fig. 4 under the Java application) running until the button is released. The operator can choose whether to send a movement signal to the robot for the selected axis and for that amount of time, or to change the time/axis before sending the actual movement signal, only after releasing the button. Such a feature potentially provides the operator with greater control over its actions.

Once a movement was completed, a data feedback from the robot's encoders is presented (in the middle of Fig. 4) to the operator, enabling him to compare the actual positioning of the robotic arm, to the one predicted by the virtual simulation.

#### 4.2.2 The Visual Feedback Module

Two video feeds, as seen in the right hand side of Fig. 4, are available for the user. These feeds provide the user with two different viewing angles of the workstation: an isometric overall view (the lower feed), and a zoomed top view (the upper feed). The top view also enables the user to snap a picture of the work area and to save it in his folder on the server. This option is introduced in order to support the maintenance of lab reports by future users of the system.

#### 4.2.3 The Data Interaction Module

The bottom left side of Fig. 4 shows the data interaction module serving three purposes: 1) uploading files to the server, 2) viewing the personal folder of the logged user, and 3) running simulation files stored in this folder. This module is not used during this research and was extensively described in Goldstain et al. [1].

### 4.3 The Subjects Group

The subject group for the conducted experiments were senior (fourth year) students, at the Computer Integrated Manufacturing Laboratory in the Industrial Engineering Department at Tel-Aviv University.

The subjects' age range lie between the ages of 20 to 30. The gender distribution was 53 percent males and 47 percent females. All subjects had a technical background resulting from their engineering education. Overall, 126 experiments were conducted, throughout five semesters.

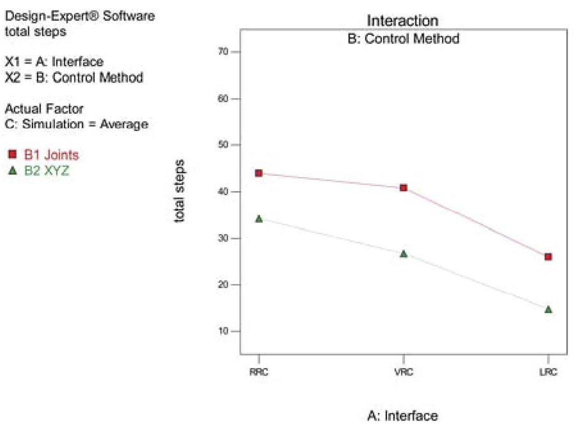


Fig. 5. Total number of steps for different interfaces and two control methods. Red squares represent the Joints control method and green triangles represent the axis-XYZ control method.

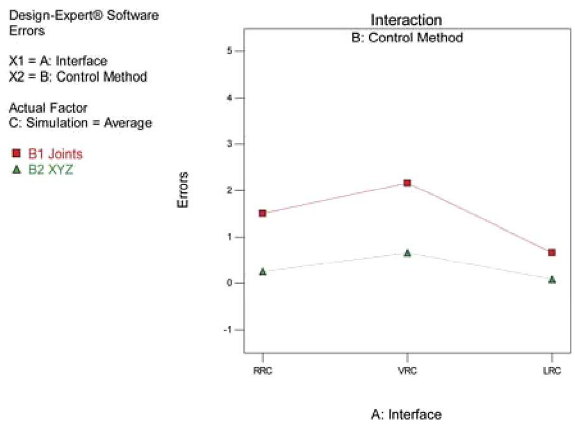


Fig. 6. Number of errors for different interfaces and two control methods. Red squares represent the Joints control method and green triangles represent the axis-XYZ control method.

The selection of this field of subject is obvious, as engineering students are most likely the target end users of any system that might be developed, based on the results of this work.

## 5 Experimental Results

### 5.1 Findings from Phase #1 of the Experiments

Phase #1 of the experiments focused on the evaluation of the three Robocell interfaces: the local Robocell interface, the remote Robocell interface that includes the VRTP (VRC), and the remote Robocell interface without the VRTP (RRC). As indicated in Chapter 3, the evaluation focused on the XYZ versus the Joints control method, and on the contribution of the preliminary simulation tool prior to the execution of the task.

#### *5.1.1 The Total Number of Required Steps*

The total number of steps required to complete the task was measured from the homing point (the location of the gripper after the calibration of the robot) until the completion of the entire task, i.e., after marking the third circle. This measure provided us with information and an intuitive understanding regarding the complexity level of the used interface.

Fig. 5 presents the average total number of steps required in each of the three Robocell-based interfaces, (starting from the RRC on the left hand side of the chart, continuing with the VRC in the middle and ending with the LRC on the right) and for each of the control methods. First, we note that as the interface approaches a realistic local setting, the number of steps decline. One can clearly see that the Joints control method results in a significantly larger number of steps compared to the axis-XYZ control method. This trend is consistent through all the tested interfaces. This result is quite intuitive, as the Axis-XYZ control method is less complex and is more intuitive (Chapter 3.1).

Examining the influence of the preliminary simulation tool on the total number of steps required, one could see that the preliminary simulation often leads to lower number of steps required.

#### *5.1.2 Number of Errors during the Tasks' Execution*

The number of errors was measured by counting the number of marks made by the operator outside the designated circles in the working area. As seen for the previous factor, this factor gives us some indication regarding the complexity of the task and points to the expertise improvement of the operator while completing the task. We expect the number of errors to be lower in designs that support better learning, as the user is expected to adapt faster to better control over the system and therefore to perform more accurately and with fewer errors.

In Fig. 6, we see the effect of the control method used by the operator on the number of errors. As expected, the axis-XYZ control method leads to a significantly lower number of errors with respect to the Joints control method and for all the examined interfaces in this phase.

Surprisingly, the highest number of errors was obtained in the VRC interface rather than the RRC interface (the one without the VRTP). This result might be explained by the attractiveness of the VRTP feature, causing the user to rely on the virtual feedback even when it causes errors. The virtual feedback (VRTP), as informative as it is, is not as accurate as the online video feedback. When using the RRC interface without the virtual feedback, the operator had to wait for the video buffering delay to end, and therefore every movement took a longer time to finish. Accordingly, the operator gave a higher attention to each robotic move before actually executing it. We believe that this extra time and attention lead to fewer errors.

Note that for the local-LRC interface, the advantage of the axis-XYZ with respect to Joints is smaller than the advantage in the remote interfaces. In fact, for the LRC interface the control intervals overlap in contrast to the other interfaces. This fact emphasizes the importance of the control method when designing a remote interface.

Figs. 7 and 8 explore the interaction between the control method and the preliminary simulation module, when using the axis-XYZ or the Joints control, respectively.

The graphs in Figs. 7 and 8 present again the average number of execution errors. This time with red squares representing a setting without a preliminary simulation,

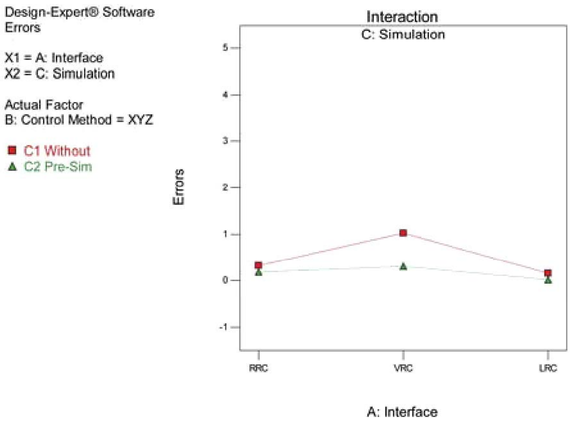


Fig. 7. Number of errors for different interfaces using axis-XYZ control method. Red squares represent the using preliminary simulation and green triangles represent no simulation.

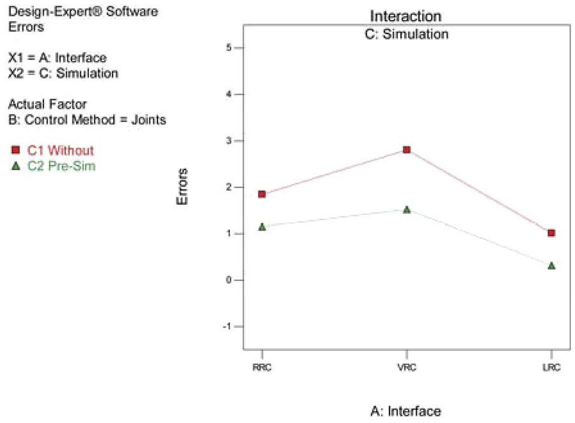


Fig. 8. Number of errors for different interfaces using Joints control method. Green triangles represent the using preliminary simulation and red squares represent no simulation.

and with green triangles representing a setting with the use of a preliminary simulation.

While using the axis-XYZ control method (Fig. 7), the average number of execution errors for both the local LRC interface and the remote RRC interface (without the VRTP) are found to be almost indistinguishable, regardless of the use or the absence of the preliminary simulation module. In the VRC interface, on the other hand, a lower number of execution errors is obtained when implementing a preliminary simulation.

Note that for the Joints control method, the use of preliminary simulation results in a significantly lower (better) average number of errors, in comparison to a situation without the preliminary simulation. This observation is consistent throughout all three different interfaces.

In both Figs. 7 and 8 one can see that the contribution of the preliminary simulation tool to decrease the number of execution errors is effected by the chosen control method. Preliminary simulation results in better execution (hence, better learning) when using an interface operated by the Joints control method.

The effect of the preliminary simulation on the number of errors while using the VRC interface seems to be almost indifferent to the control method in use. In this case, preliminary simulation results in a lower number of errors for both control methods, speculating that the VRTP module reduces the complexity gap between axis-XYZ and Joint control methods.

### *5.1.3 Learning and Improvement Measures*

The learning and improvement rate, as defined in Chapter 3.1, is calculated by dividing the total number of steps that are required to perform the second and third legs, by the total number of steps required to complete the task. Such a measure is expected to be lower when a better learning curve exists. A lower measure will indicate better learning as it represents a significantly higher number of movements for the first leg, and therefore a greater margin between the legs. This will implicate improved performance and an efficient learning process.

In Fig. 9, we see the average learning function calculated for all three different interfaces during phase #1.

When using the local LRC interface, the Joints control method result in a better (lower) learning factor. This result is explained by the simplicity of the task when performed locally and with the simplest control method (axis-XYZ). A simple task leaves very little room for improvement as its execution requires almost the minimal number of steps possible, already from the first leg.

When using the remote interfaces (RRC and VRC), the axis-XYZ control method results in slightly better improvement rate than the Joints control method.

The best improvement and learning rate is obtained for the RRC interface, either with the axis-XYZ or with Joints control method. This result can also be explained by the complexity of the task, which is the highest in these settings, leaving much room for learning and improvement. These observations results from the fact that the main difficulty in performing the complex tasks is mental, while the simplest tasks are associated with mechanical difficulty. Thus, a higher potential for improvement is related to the former.

## 5.2 Findings from Phase #2 of the Experiments

Phase #2 of the experiment focuses on the evaluation of the differences between the web-based remote interface (INTERNET) and the three Robocell-based interfaces: the

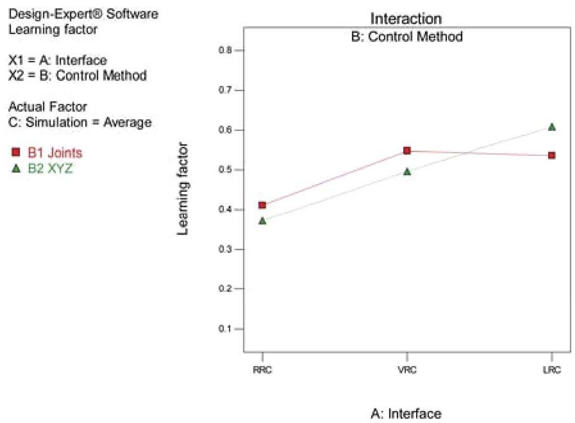


Fig. 9. Improvement rate for different interfaces and two control methods. Red squares represent the Joints control method and green triangles represent the axis-XYZ control method.

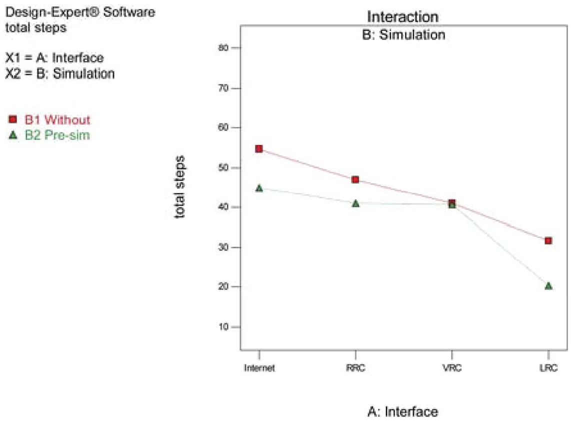


Fig. 10. Total number of steps for different interfaces using Joints control method. Green triangles represent the using preliminary simulation and red squares represent no simulation.

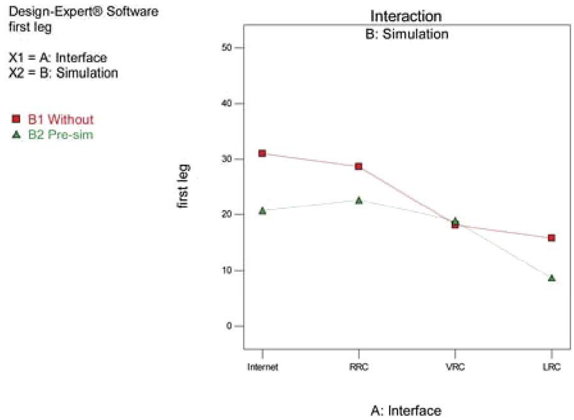


Fig. 12. Number of steps in first leg for different interfaces using Joints control method. Green triangles represent the using preliminary simulation and red squares represent no simulation.

local settings (LRC), the remote settings with the VRTP module (VRC), and the remote settings without the VRTP (RRC). All four interface settings were operated with the Joints control method to evaluate the effect of the preliminary simulation tool, prior to the online execution of the task.

### 5.2.1 The Total Average Number of Required Steps

Fig. 10 analyses the interaction between the interface type and the use of a preliminary simulation tool. When considering the three remote interfaces (the three most left interfaces in the abscissa), we see that the effect of the preliminary simulation to decrease the average total number of steps is most significant for the INTERNET interface. Such an effect barely exists in the VRC interface. Yet, the preliminary simulation also affects the total number of steps when using the local LRC interface.

Out of these three remote interfaces, the VRC interface is surprisingly indifferent to a preliminary simulation (as observed in phase #1 of the experiment), and results in the same average number of steps both with and without using the preliminary simulation. This result may be explained by the fact that using the advanced VRTP tool during the task execution makes the preliminary simulation, which is based on the same tool, redundant.

### 5.2.2 The Number of Occurred Errors

Fig. 11 presents the average number of execution errors for each of the four interfaces with or without the preliminary simulation tool. One can see that when using the preliminary simulation (marked by green triangles) the difference between the three remote interfaces is negligible (less than 0.5 errors in average).

Moreover, when the preliminary simulation is not used (marked by red squares) the remote RRC interface leads to less errors than the other remote interfaces, obtaining an error rate which is close to the one obtained by the local LRC benchmark interface.

### 5.2.3 Learning and Improvement Measured

Next, we present the improvement in the learning rate of the users. We depict the learning rate for each of the legs as a function of the interface type and the use of a preliminary simulation—two design factors that the interaction between them is found to be statistically insignificant (a P-Value of 0.22).

Figs. 12 and 13 present the number of steps required for the first leg (marked as “H-to-1”) and for the “latest” legs (marked as “1-to-3”), respectively. The results in each graph represent separately the use (by red squares) or lack of use (by green triangles) of the preliminary simulation.

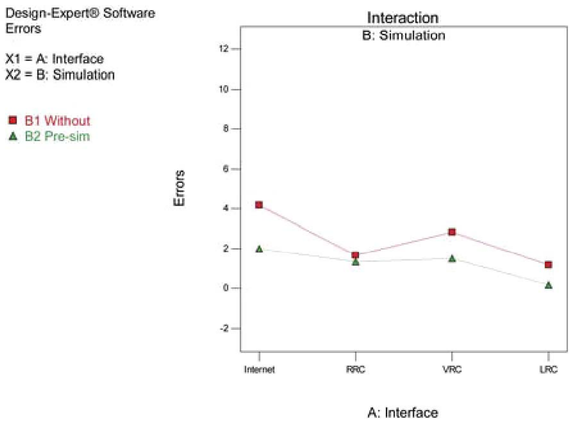


Fig. 11. Number of errors for different interfaces using Joints control method. Green triangles represent the using preliminary simulation and red squares represent no simulation.

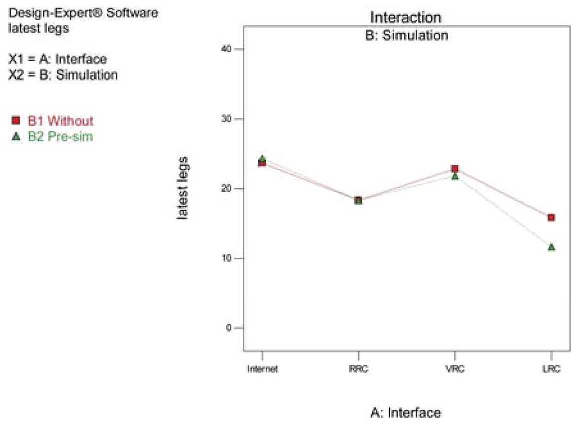


Fig. 13. Number of steps in latest legs for different interfaces using Joints control method. Green triangles represent the using preliminary simulation and red squares represent no simulation.

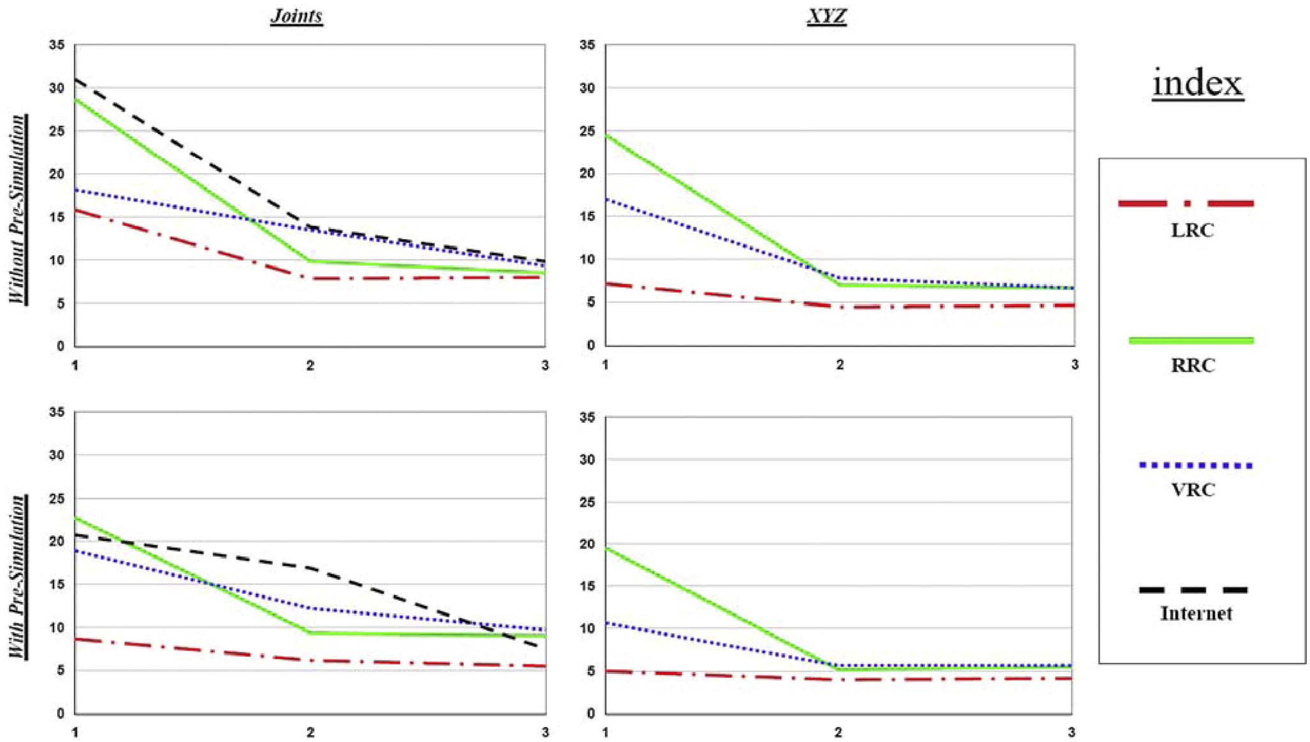


Fig. 14. Learning curves for the experiments. The horizontal axis represents the three sections (legs) of the experiments, starting from the homing point at the left hand side, through to the end of the execution on the right. The vertical axis shows the number of steps required for the specific leg of the execution. Red curves represent the local LRC test interface. Green curves represent the Remote RRC interface. Blue curves represent the VRC interface. Black curves represent the Internet interface. The right-hand side graphs are showing three different curves differentiated by the control method used, and by the chosen interface (Internet interface excluded). The left-hand side graphs are showing four different curves differentiated by the use or absence of preliminary simulation, and the by the chosen interface (all four types of interfaces are included).

As seen from the total number of required steps (Fig. 10) and from the learning graphs (Fig. 14), the use of a preliminary simulation results in fewer steps required to complete the first leg for all interfaces except the VRC interface. From Fig. 13, we see that the number of steps required to complete the latest two legs is indifferent to the use of a preliminary simulation, i.e., resulting in roughly similar performance measured both with and without the use of the preliminary simulation tool prior to the execution.

These results suggest that the ability to simulate movements in the cell before an actual execution provides the operator with an early stage of learning, which results in an improved performance at the beginning of the execution. The effect of such learning diminishes in later stages.

Another appearing result of these two graphs is that when a VRTP module is available, the contribution of the preliminary simulation is limited.

## 5.3 Learning Curves for Different Designs

The graphs shown in Fig. 14 emulate the learning curves of the task execution, for four combinations of the examined factors.

### 5.3.1 Comparison within Control-Method Factor

The two graphs on the right side of Fig. 14 present the results of phase #1 of the experiments.

In each examined interface, both the Joint and the axis-XYZ control methods result in very similar slopes of the learning curve, differing only in the height from the abscissa. These differences result from the heights of start points, which are lower for the (more intuitive) axis-XYZ control method. The improvement throughout the steps seems to be almost unaffected by the actual control method in use. However, we see that while the end points of all curves are bounded by a narrow margin between 4 and 10 steps that are required to complete the last leg, the curves have greater margin at their starting points. This result leads to the conclusion that the actual improvement of execution achieved during the task is greater in the RRC interface (where the user is located far from the station) and the lowest in the local LRC benchmark interface (where the user is close to the station).

### 5.3.2 Analyzing the Preliminary-Simulation Factor

The two graphs on the left side of Fig. 14 present the results of phase #2 of the experiments.

Unlike the right side graphs, the learning-curve slopes in each interface are different here when compared with or without a preliminary simulation. For both the RRC and the local LRC interfaces, we observe steeper curves for experiments without preliminary simulation, indicating that the actual improvement of execution achieved during the task is greater when operating the system without preliminary simulation. Nevertheless, we note that the curves of each interface end closely to each other, as the users reach the same average number of steps in the last leg of the execution, and regardless of the use of a preliminary simulation. This observation leads to conclude that preliminary simulation supports the learning prior to tasks' execution, and therefore this module is recommended for use.

The above conclusions are also supported by the learning curves for the Internet interface. As we can see for this more difficult working interface, the difference between the curves is noticeable suggesting that when using a preliminary simulation, most of the learning occurred during the simulation part, leaving very little room for improvement during the actual execution of the tasks.

The curves for the VRC interface support our assumption that the VRTP tool and the preliminary simulation are superfluous to each other. This is supported by the almost identical curves (both in slope and in height) indicating indifference to the presence or absence of the preliminary simulation in the process.

## 6 Discussion

### 6.1 Analysis of the Results

Let us start by addressing the impact of the chosen control method on the learning outputs. The results clearly show that the axis-XYZ control method leads to a smaller number of steps and a lower number of errors for all considered interfaces, including the Internet interface. The research shows that the chosen control method has an impact on the learning curve as a result of the different complexity related with each control method.

The preliminary simulation module was found to be an effective learning tool prior to the execution of a task in an online environment. Its effects are more significant in higher complexity settings, when using the Joints control method. Nevertheless, the preliminary simulation is found to be superfluous when using the VRTP module, in terms of minimizing the number of steps or improving the operators' learning.

Learning curves of operators that used a preliminary simulation tool show less improvement during the online measured execution, although these curves resulted in approximately the same performance at the end. These results indicate that a major part of the learning process happens during the preliminary simulation stage (for the Internet interface in particular), providing a better start point to the operators at the online stage.

When analyzing the number of errors measured both in phase #1 and in phase #2 of the experiments, it was found that when using preliminary simulation the difference between remote interfaces is negligible, as all remote interfaces lead roughly the same average number of errors. However, when a preliminary simulation tool is unavailable, the RRC interface provides the best results in terms of minimizing the number of errors out of all the considered remote interfaces. The RRC interface was found to be indifferent to the use of a preliminary simulation tool when using the axis-XYZ control method. Moreover, the RRC interface, along with the Joints control method, resulted in the lower number of errors with respect to the VRC, both with and without a preliminary simulation.

Although the Internet interface is the most complex interface among the considered ones, when it is combined with a preliminary simulation tool, it provides almost as good results as the rest of the remote interfaces (in terms of the number of steps required). The same conclusion is drawn with respect to the number of errors. We believe that the ability to reconsider a movement once it has been chosen, yet before it is executed, was a significant factor in explaining the success of this interface.

#### *6.1.1 Conclusion and Guidelines for Design*

When required to design a remote teleoperation interface, we need to choose the appropriate combination of components in order to meet our learning/teaching goals. If the goal is, as in our study, an accurate operation, then the suggested control method should be the axis-XYZ control, as it leads to a lower number of errors. However, since sometimes part of our robot operation teaching would benefit from teaching alternative control methods, and as it seems that selecting either control method will not affect the achieved learning rate of the user, it is suggested to have both control methods available if and whenever it is possible.

A preliminary simulation module is highly suggested on the design of a remote telerobotic interface. The only module that was found to have the same impact as the preliminary simulation tool in terms of improving the operators learning and performance was the VRTP module. The VRTP module provides the user with the same learning qualities as the preliminary simulation tool, but this time during the actual online work. If the designed system has to service a large number of users, by relying on short online time windows for each user, then a preliminary simulation is the most effective tool for learning. However, if one can provide each user with enough online access time to the robotic cell, then it is recommended to integrate a VRTP module into the interface.

A main feature that differs in the Internet interface from the other considered interfaces was the ability to reconsider a movement prior to its execution. We believe that this feature affected the higher learning rate found for this interface, and recommend facilitating such mechanism into future designs of remote telerobotic interfaces.

### 6.2 Further Research

Further research in this field can address the affect of integrating a VRTP tool and an axis-XYZ control method into an Internet interface. Results drawn from this research suggest that such integration can yield the best remote learning performance.

Another research could focus on visual aspects associated with remote telerobotic learning, examining positioning, and orientation of cameras and their effect on the user's comprehension of the three-dimensional work area, as well as on his learning performance.

In relation to teaching laboratories, useful work can be done for designing remote-compatible tasks for learning robotics, as not all available routines for teaching robotics are applicable for remote learning without the presence of an instructor on the site.

## Acknowledgments

The authors would like to thank the research fund of the Israel Internet Association for partially funding this research. They would also like to thank the staff of the CIM Laboratory for helping throughout this research.

## References

[1] O. Goldstein, I. Ben-Gal, and Y. Bukchin, "Remote Learning for the Manipulation and Control of Robotic Cells," *The European J. Eng. Education*, vol. 32, no. 4, pp. 481-494, 2007.

[2] J. Bukchin, R. Luquer, and A. Stubs, "Learning in Tele-Operations," *IIE Trans.*, vol. 34, pp. 245-252, 2002.
[3] R. Siegwart and P. Saucy, "Interacting Mobile Robots on the Web," *Proc. IEEE Int'l Conf. Robotics and Automation*, 1999.
[4] NASA Space Telerobotics Program, http://rainer.oact.hq.nasa.gov/telerobotics_page/telerobotics.shtml, 2011.
[5] I. Belousov, J. Tan, and G. Clapworthy, "Teleoperation and Java3D Visualization of a Robot Manipulator over the World Wide Web," *Proc. Information Visualisation*, pp. 543-548, July 1999.
[6] I. Belousov, R. Chellaliand, and J. Clapworthy, "Virtual Reality Tools for Intemet Robotics," *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA '01)*, 2001.
[7] M. Wang and J.N.K. Liu, "A Novel Teleoperation Paradigm for Human-Robot Interaction," *Proc. IEEE Int'l Conf. Robotics, Automation and Mechatronics*, pp. 13-18, 2004.
[8] H.S. Hu, L.X. Yu, P.W. Tsui, and Q. Zhou, "Internet-Based Robotic Systems for Teleoperation," *Assembly Automation*, vol. 21, pp. 143-151, 2001.
[9] J. Kofman, W. Xianghai, T. Luu, and S. Verma, "Teleoperation of a Robot Manipulator Using a Vision-Based Human-Robot Interface," *IEEE Trans. Industrial Electron*, vol. 52, no. 5, pp. 1206-1219, Oct. 2005.
[10] H. Hu Yu, L.P.W. Tsui, and Q. Zhou, "Internet-Based Robotic Systems for Teleoperation," *Assembly Automation*, vol. 21, no. 2, pp. 143-151, 2001.
[11] T. Ravindra, L.H. Huang, L. Shi-Jer, and R. Diez, "Controlling Robot through Internet Using Java," *J. Industrial Technology*, vol. 20, no. 3, 2004.
[12] C. Röhrig and A. Jochheim, "Java-Based Framework for Remote Access to Laboratory Experiments," *Proc. IFAC/IEEE Symp. Advances in Control Education*, 2000.
[13] S. Yang, H.X. Zuo, and L. Yang, "Controlling an Internet-Enabled Arm Robot in an Open Control Laboratory," *Assembly Automation*, vol. 24, no. 3, pp. 280-288, 2004.
[14] S.H. Chen, R. Chen, V. Ramakrishnan, S.Y. Hu, Y.K.C.C. Zhuang, and B.M. Chen, "Development of Remote Laboratory Experimentation through Internet," *Proc. Hong Kong Symp. Robotics and Control*, vol. 2, pp. 756-760, 1999.
[15] A. Bischoff and C. Röhrig, "Remote Experimentation in a Collaborative Virtual Environment," *Proc. 20th World Conf. Open Learning and Distance Education*, Apr. 2001.
[16] D.W. Calkin, R.M. Parkin, R. Safaric, and C.A. Czarnecki, "Visualization, Simulation and Control of a Robotic System Using Internet Technology," *Proc. Fifth IEEE Int'l Advanced Motion Control Workshop*, pp. 339-404, 1998.
[17] C. Fernandez, M.A. Vicente, and L.M. Jimenez, "Virtual Laboratories for Control Education: A Combined Methodology," *Int'l J. Eng. Education*, vol. 21, no. 6, pp. 1059-1067, 2005.
[18] D. Lowe, S. Murray, E. Lindsay, and D. Liu, "Evolving Remote Laboratory Architectures to Leverage Emerging Internet Technologies," *IEEE Trans. Learning Technologies*, vol. 2, no. 4, pp. 289-294, Oct. 2009.
[19] F.A. Candelas, S.T. Puente, F. Torres, V. Segarra, and J. Navarrete, "Flexible System for Simulating and Tele-Operating Robots through the Internet," *J. Robotic Systems*, vol. 22, no. 3, pp. 157-166, 2005.
[20] F. Michau, S. Gentil, and M. Barrault, "Expected Benefits of Web-Based Learning for Engineering Education: Examples in Control Engineering," *European J. Eng. Education*, vol. 26, no. 2, pp. 151-168, 2001.
[21] W.A. Khan, S.M.R. Al-Doussari, and A.H.M. Al-Kahtani, "Establishment of Engineering Laboratories for Undergraduate and Postgraduate Studies," *European J. Eng. Education*, vol. 27, no. 4, pp. 425-435, 2002.
[22] A.S.C. Fernandes and M.J. Martins, "Self-Training through the Internet," *European J. Eng. Education*, vol. 26, no. 2, pp. 169-177, 2001.
[23] S.T. Puente, F. Torres, F.G. Ortiz, F.A. Candelas, "Remote Robot Execution through WWW Simulation," *Proc. 15th Int'l Conf. Pattern Recognition*, 2000.
[24] X. Yang, Q. Chen, D.C. Petriu, and E.M. Petriu, "Internet-Based Teleoperation of a Robot Manipulator for Education," *Proc. Third IEEE Int'l Workshop Haptic, Audio and Visual Environments and Their Applications*, pp. 7-11, 2004.
[25] L.S. Enrique, J. Noguez, and G. Huesca, "Project Oriented Learning for Basic Robotics Using Virtual Laboratories and Intelligent Tutors," *Proc. 35th ASEE/IEEE Frontiers in Education Conf.*, Oct. 2005.
[26] R. Marin, P.J. Sanz, P. Nebot, and R. Wirz, "A Multimodal Interface to Control a Robot Arm via the Web: A Case Study on Remote Programming," *IEEE Trans. Industrial Electronics*, vol. 52, no. 6, pp. 1506-1520, Dec. 2005.
[27] M. Cooper and J.M.M. Ferreira, "Remote Laboratories Extending Access to Science and Engineering Curricular," *IEEE Trans. Learning Technologies*, vol. 2, no. 4, pp. 342-353, Oct. 2009.
[28] C.S. Tzafestas, N. Palaiologou, and M. Alifragis, "Virtual and Remote Robotic Laboratory: Comparative Experimental Evaluation," *IEEE Trans. Education*, vol. 49, no. 3, pp. 360-369, Aug. 2006.
[29] A. Eliav, T. Lavie, Y. Parmet, H. Stern, J. Waches, and Y. Edan, "Kiss Human-Robot Interfaces," *Proc. 18th Int'l Conf. Production Research*, 2005.
[30] B. Aktan, C.A. Bohus, L.A. Crowl, and M.H. Shor, "Distance Learning Applied to Control Engineering Laboratories," *IEEE Trans. Education*, vol. 39, no. 3, pp. 320-326, Aug. 1996.
[31] M.A. Bochiccio and A. Longo, "Hands-On Remote Labs: Collaborative Web Laboratories as a Case Study for IT Engineering Classes," *IEEE Trans. Learning Technologies*, vol. 2, no. 4, pp. 320-330, Oct. 2009.
[32] J.A. Adams, "Critical Considerations for Human-Robot Interface Development," *Proc. AAAI Fall Symp. Human-Robot Interaction*, pp. 1-8, 2002.
[33] Tel-Aviv Univ. CIM-Lab, "Remote Learning for Robotic Cells," http://www.eng.tau.ac.il/remote, 2011.

**Ofir H. Goldstain** received the BSc (2004) and MSc (2009) degrees in industrial engineering from Tel-Aviv University and is currently a doctoral student in the Department of Industrial Engineering. He has been a researcher and an instructor in the Computer Integrated Manufacturing Laboratory for the last six years. His interests include remote learning, robotics, vision systems, and human-factors engineering.

**Irad Ben-Gal** received the BSc (1992) degree from Tel-Aviv University, and the MSc (1996) and PhD (1998) degrees from Boston University. He is an associate professor at Tel-Aviv University, where he is the head of the Computer Integrated Manufacturing Laboratory. 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 boards of several journals. His papers have been published in *IIE Transactions*, the *International Journal of Production Research*, *Technometrics*, the *IEEE Transactions on Bioinformatics*, and other journals. He has 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. He is a member of the IEEE.

**Yossi Bukchin** received the BSc, MSc, and DSc degrees in industrial engineering and management from the Technion Israel Institute of Technology. He is an associate professor in the Department of Industrial Engineering at Tel Aviv University. He is a member of the Institute of Industrial Engineering (IIE) and the College-Industry Council on Material Handling Education (CICMHE) and is on the editorial board of *IIE Transactions*. He has held a visiting position in the Grado Department of Industrial & Systems Engineering at Virginia Tech. His papers have been published in *IIE Transactions*, *Operations Research*, *SOM*, the *European Journal of Operational Research*, the *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, multiobjective optimization, and work station design with respect to cognitive and physical aspects of the human operator.