Distance Learning in Robotics and Automation by Remote Control of Lego Mobile Robots

Fabio Carusi, Marco Casini, Domenico Prattichizzo, Antonio Vicino
Dipartimento di Ingegneria dell'Informazione
Università di Siena
via Roma 56, 53100 Siena, Italy
Email: {carusi,casini,prattichizzo,vicino}@ing.unisi.it

Abstract— This paper presents the Automatic Control Telelab (ACT), a remote laboratory for education in robotics and automation developed in Siena. A special attention will be devoted to the new ACT process, a Lego mobile robot, which has been recently added to the telelab. Remote users can control the Lego mobile robot, and all the other processes, using an Internet browser. Moreover users can design new controllers by means of the Matlab/Simulink environment. The ACT home page is http://www.dii.unisi.it/∼control/act.

I. INTRODUCTION

Internet and Web technologies are changing dramatically the education in robotics and automation. Many distance learning paradigms have been developed in recent years and among these, laboratories accessible through the Internet are certainly the most effective ones. Consider for instance the tele-laboratory developed in [1] or the project described in [2] where an expensive resource, such as a mobile robot, is shared on the web. Regarding the web technologies used in robotics and automation courses, exhaustive contributions have been provided in [3], [4] where the authors describe the use of virtual [5] and remote labs in control teaching. A distinguished feature of remote labs with respect to virtual labs is that users can interact with real physical processes through the Internet. The implementation of a remote lab requires in general more efforts. Moreover, the complexity of the hardware and software architecture design increases when the remote lab allows the user to design the controller within the remote web session. This is for instance the case of the remote lab developed at the Oregon State University [6], where students can remotely control a robot arm not only changing some parameters but, more interestingly, transmitting the control program which changes the dynamics of the closed-loop system.

A remote laboratory can either use a well–known software environment, such as LabVIEW [7] and Matlab/Simulink [8], [9], [10], or use some special purpose software, as in [1], [11]. It is the authors' opinion that well–known software environments are better suited to spread out the remote laboratory practice. Usually, students do not want to learn ad–hoc control languages which are tailored for the particular remote lab and they like to take advantage of control functions developed in other well–known contexts. On the other side, integrating a well–known software interface, like Matlab/Simulink, in the remote lab architecture is a difficult task and makes the overall remote lab design more complex.

The Automatic Control Telelab (ACT) has been developed in Siena since 1999 [12], [13]. It is a remote laboratory for robotics and automation education. The ACT allows the user to choose a predefined controller to steer the process, or, more interestingly, to remotely synthesize a new controller through the Matlab/Simulink environment and its toolboxes.

In this paper the ACT and a new ACT process, a Lego mobile robot developed during the Robotics class held at University of Siena in 2003, are presented. Students can design their own controller to track a user-defined trajectory other than running predefined controllers. Four webcams increase the telepresence during the remote experiment. The new Lego Mindstorms Robotics Invention System [14] is a good example of low-cost platforms for teaching integrated systems in robotics education [15]. A nice piece of work involving Lego Mindstorms and robotic applications is that discussed in [16] where authors present a very interesting education project on surgical robotics for college students.

One of the main features of the ACT project consists in simplifying the user interface in a way that only the very basic notions of Simulink should be known in order to design the controller of the Lego mobile robot, as far as the other experiments. During the experiment it is also possible to change some typical controller parameters and the reference signal.

II. THE AUTOMATIC CONTROL TELELAB

The Automatic Control Telelab is a remote laboratory mainly intended for educational purpose, and since 1999 it has been used in control systems classes [12], [13].

The aim of the project is to allow students to easily put in practice their theoretical knowledge of robotics and automation without restrictions due to laboratory opening time and processes availability. The ACT is accessible 24 hours a day from any computer connected to the Internet by means of any common browser. No special software or plug-in is required. Users can design new controllers by means of the Matlab/Simulink environment. However, if Matlab is not available on the local computer, the user can still run experiments by choosing among the available predefined controllers. A live video window is provided for each remote experiment session.

Fig. 1.   The other Automatic Control Telelab's on–line experiments.

At present, in addition to the Lego mobile robot which will be extensively described in the following sections, four other processes are available for remote control: a DC motor, a tank for level control, a magnetic levitation system and a two degree–of–freedom helicopter (Fig. 1). The DC motor is used to control the angular position and velocity of a rotating axis. The level control process has been included because, in spite of its simplicity, it shows nonlinear dynamics, whereas the magnetic levitation process, being nonlinear and unstable, exhibits very interesting properties to be analyzed in control theory education. Finally, the two degree–of–freedom helicopter, which is a nonlinear and unstable MIMO system, can be used by graduate students.

One of the main features of the ACT is the possibility of integrating any user-defined controller in the control loop of the remote process. The controller synthesis is based on the Matlab/Simulink environment. Since Matlab and Simulink packages are standard tools in control systems classes, there are no additional hurdles for a student who wants to design new controllers. In fact, a controller is just a Simulink model, similar to those commonly used to run simulations of dynamic systems.

Other than control experiments, the ACT also allows to perform *remote system identification* experiments [17].

An interesting feature of the ACT is the *student competition* facility. A typical remote laboratory allows students to run an experiment and see the dynamic response, but in general no information on controller performances is provided and it is not possible to know how controllers designed by other people behave on the same process. This is one of the reasons that motivated us to design a student competition mechanism for our remote laboratory. Through this tool a student knows about performance requirements his/her own controller must satisfy. So, in addition to standard controller synthesis exercises, this tool allows a student to design a controller, which must satisfy some performance requirements, and to test it on remote real processes. At the end of the experiment, the performance

indexes are automatically computed and shown to the user. For a thorough treatment on the student competition feature the reader is referred to [18].

III. EXPERIMENT DESCRIPTION

The aim of this project is to allow students to perform remote control experiments on a mobile robot through the Internet. In particular, the robot is made of Lego Mindstorms bricks [14], and is a part of the ACT.

The control of a mobile robot needs special features, such as security systems to prevent the robot to go out the working area, an automatic recharge system, a mechanism to detect the robot position, and so on. All these aspects are described in detail in Section IV.

Through this facility, students have to design a control law which allows the robot to follow a given trajectory. The control algorithm is implemented as a Simulink model, giving students an easy and powerful tool for controller design. The trajectory can be designed by the user or chosen among a set of predefined ones.

Since all aspects regarding communications and process interfacing are provided by the ACT, students may focus their efforts in designing the controller.

Predefined control algorithms as well as user-defined ones can be used to track a given path. Moreover, it is possible to set controller parameters to be tuned on-line, and to choose and change the trajectory reference during the experiment.

While the experiment is in progress, it is possible to see some signals such as the position of the robot, the required reference and the commands applied to the motors. Live videos from four cameras are also provided to increase the sense of presence in the lab.

At the end of the experiment, all the significant signals can be downloaded in Matlab, allowing a student to perform off-line analysis.

In Section VI a complete work session is described.

IV. HARDWARE DESCRIPTION

In this section the mobile robot and other hardware components involved in this project are described.

A. The mobile robot

The mobile robot (Fig. 2) has been built with Lego Mindstorms bricks [14]. Lego Mindstorms allows the use of the so-called RCX, a special brick which contains a microprocessor and which allows to be connected to three sensors (e.g. encoders, light or touch sensors) and three actuators (usually DC motors). Moreover, it is able to store and run up to four programs. The RCX communicates with a PC through a serial or a USB infrared tower, by means of which it is possible to download a program and to perform other basic operations.

In this project, the RCX is not used in the classical way (uploading a program and running it), but it is only used to communicate and execute commands provided by the PC. In other words, the control algorithm does not run on the RCX but on the PC. The RCX only drives motors accordingly to the
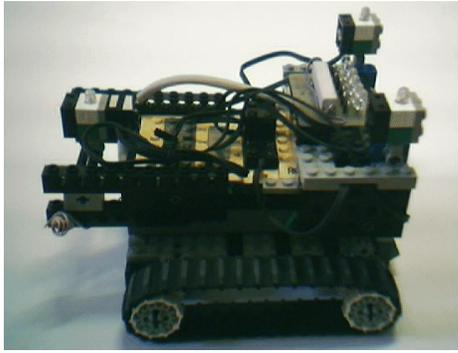
Fig. 2.   The mobile robot built with Lego Mindstorms.

Fig. 3.   The box for automatic recharge of the robot.

Fig. 4.   Actual and filtered images used to detect the robot.

received signals. This allows a user to overcome the limitations in terms of memory and program complexity imposed by the RCX, allowing a user to design complex control laws.

The robot is a two tracks vehicle. Tracks are driven by two Lego DC motors whose speed is reduced by a factor of 9 by means of an appropriate cogwheels system in order to increase torque as well as decrease speed. The input signals to Lego motors are quantized between -8 to 8 in integer steps.

Three lights have been put on the top of the robot. These lights are disposed as an isosceles triangle in order to easily detect the position and orientation of the robot by means of a camera placed above the working area (see Subsection IV-B).

The robot is supplied by 6 rechargeable alkaline batteries (AA type) which provide an overall voltage of 9 volts. To allow an automatic recharge at the end of every experiment, a special circuit has been built on the robot, so that batteries can be recharged when the robot goes inside the recharge box (Fig. 3).

B. The detection system

Since proprioceptive sensors (i.e. encoders) would not be sufficient to detect the robot position with a high level of accuracy, a camera has been put above the working area with the purpose to acquire the whole area and to allow the estimation of robot position and orientation.

To perform estimation of the robot location, acquired data are first filtered in order to emphasize the lighter pixels given by the three lights placed on the robot (Fig. 4). Then, the

position and the angle of the robot w.r.t. a fixed coordinate system are computed by a fast and efficient algorithm. To prevent possible errors, environmental light is maintained constant.

The camera has been placed at an height of about 3 meters above the ground so that the overall area captured by the detection camera is of 320×240 centimeters.

Due to the high feasibility of the detection system, it is not needed that the ground of the working area be made of special material or color.

C. Safety mechanisms

Since experiments will be executed at any time, and in particular when nobody supervises the lab, security systems are essential to allow a safe and reliable experiment. There are two main safety mechanisms: the former stops the experiment whenever the robot moves outside an area of 280×200 centimeters. This prevents the robot to go out the area captured by the detection camera. The latter guarantees a maximum time after which the experiment is stopped. This avoid a user to monopolize the process. Once the experiment is finished, a special routine which allows the robot to go to the recharging box is run.

D. The user webcams

It is well known that a remote lab which also provides a live video (in addition to experiment signals) is more attracting for users, since the sense of presence in the lab is increased. For this reason, four webcams are provided to show overall and restricted views of the working area (Fig. 5). One is placed
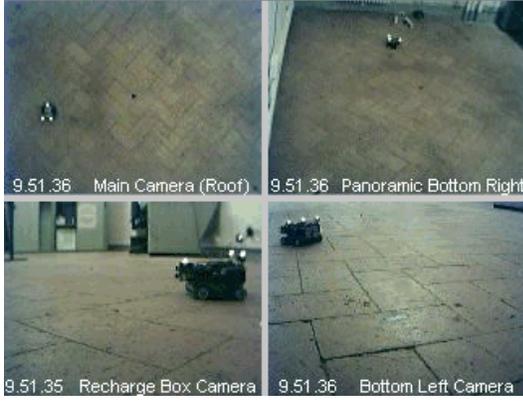
Fig. 5. The live video images from the four cameras.

Fig. 6. The interpolated experimental motors characteristic.

above the ground just like the detection camera. One has been placed to give a panoramic view. One is on the recharging box and the last one is in the left-bottom angle of the area.

V. PHYSICAL MODEL

In this section a mathematical model of the mobile robot is described.

Some robot parameters, obtained both on Lego Mindstorms data sheets and on an identification analysis, are reported in Table I. Since the RCX uses integer values steps to drive the motor, they have been referred to as *units* (they do not have any physical meaning).

TABLE I

MOBILE ROBOT PARAMETERS.

Working voltage	$9\,V$
Distance barycenter-tracks	$7.9\,cm$
Max robot speed	$\simeq 4.5\,cm/sec$
Motor command range	$[-8, 8]\,Units$
Working area	$280 \times 200\,cm$

Let us denote by X, Y and θ the coordinates of the position and the angle of the robot w.r.t. a fixed reference frame. Moreover, let v_l and v_r be the left and right track speeds respectively, and let r be the distance between the robot center and the tracks. The theoretical mobile robot kinematics is simply given by

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dfrac{v_l + v_r}{2}\cos\theta \\ \dfrac{v_l + v_r}{2}\sin\theta \\ \dfrac{v_r - v_l}{2\,r} \end{bmatrix}. \quad (1)$$

To simplify the notation, dependence of variables w.r.t. time has been omitted.

Due to frictions, unmodeled dynamics and nonlinearities which affect the robot, the model reported in (1) does not appear to be very accurate.
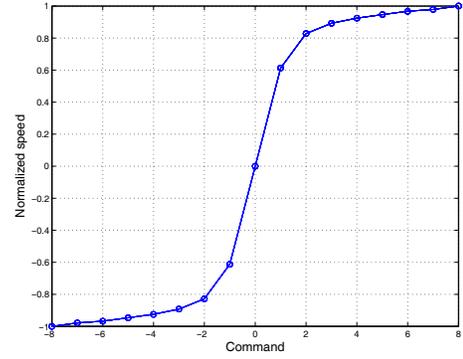
The relationship between the tracks speed and the tracks input has been experimentally estimated and is reported in Table II. Note that such a relation is strongly nonlinear (Fig. 6).

TABLE II

EXPERIMENTAL RELATIONSHIP BETWEEN COMMAND APPLIED TO MOTORS AND TRACK SPEEDS.

Track Command u_i	Normalized Speed $v_i = F(u_i)$
-8	-1
-7	-0.9785
-6	-0.9677
-5	-0.9462
-4	-0.9247
-3	-0.8925
-2	-0.828
-1	-0.6129
0	0
1	0.6129
2	0.828
3	0.8925
4	0.9247
5	0.9462
6	0.9677
7	0.9785
8	1

A better model than (1) has been experimentally obtained through a system identification procedure, giving:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 2.34\,(F(u_l) + F(u_r))\cos\theta \\ 2.34\,(F(u_l) + F(u_r))\sin\theta \\ 0.228\,(F(u_r) - F(u_l)) \end{bmatrix}, \quad (2)$$

where $F(\cdot)$ denotes the function given in Fig. 6.

VI. A SESSION EXAMPLE

From the home page of the ACT it is possible to access to general information pages, as for instance the user guide of the laboratory and the list of available processes. For every process it is possible to show a system description as well as a sample movie. In the following we refer to the Lego mobile robot.

Fig. 7.   The *Control Type Interface*.

Fig. 8.   The Simulink template model for reference and controller design.

Fig. 9.   The ACT controller subsystem.

Students have to design a controller in order to allow the robot to follow a chosen reference trajectory. They can test known control algorithms or design new ones.

Once the process has been chosen, the *Control Type Interface* shows up as in Fig. 7. Through this interface it is possible to choose the control law among a set of predefined ones. It is also possible to design a new controller; in this case, the user must have locally installed the Matlab/Simulink software, since the controller is implemented as a Simulink model. A template file (Fig. 8) may be downloaded to help the user in this step. To design a controller the user just have to open the "ACT_CONTROLLER" block and to properly connect the input signals (X, Y, and θ errors and outputs) to the output ones (left and right motor commands). It is also possible to set some parameters to be tuned on-line just naming them with the prefix "ACT_TP_" (ACT Tuning Parameter) (Fig. 9). Similarly, it is possible to choose the set of possible references, opening and modifying the "ACT_REFERENCE" block shown in Fig. 8.

Once the controller has been designed, the user can send it to the server where it is compiled and, if no errors occur, executed.

Through the *Experiment Interface* (Fig. 10), the user may choose the reference to apply, change some controller parameters and view the experiment signals. It is also possible to watch the on-line videos provided by the four cameras (Fig. 5).

The reference panel allows users to change the reference path, by choosing for instance among a step, a circle or a square trajectory. In addition to predefined references, it is also possible to draw a hand–made trajectory by selecting the "Draw Reference" checkbox; in this case, a segmented or a continuous path can be drawn.
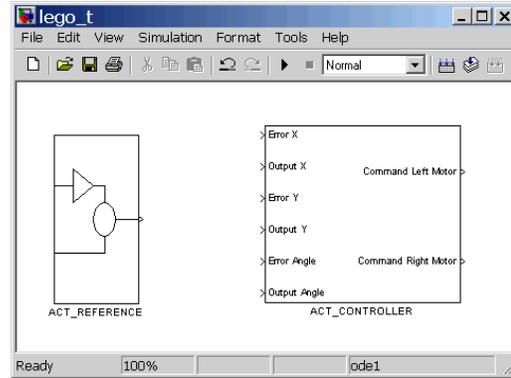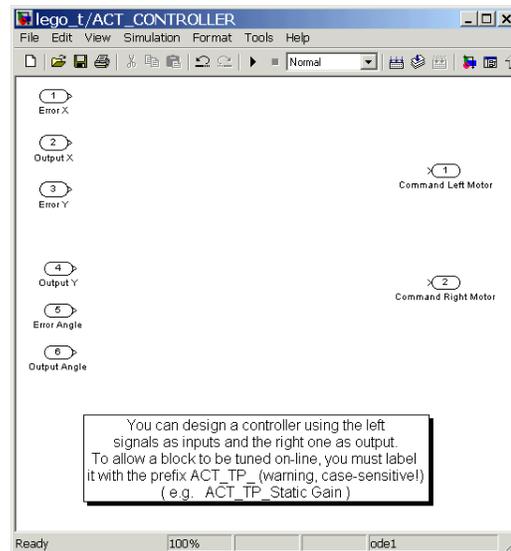
By means of the X-Y plot (Fig. 11), it is possible to on–line observe the robot position and the requested set-point. Data is updated every second and a sort of shadow increases the sense of movement of the robot. Commands to the motors are also displayable by appropriate graphics.

Once the experiment is finished, it is possible to download the full dynamics in a Matlab workspace format file (.mat). These data may be used to perform off–line analysis or to plot accurate graphics as shown in Fig. 12. It is worthwhile to stress that, after completing an experiment, the robot automatically goes into the recharging box.

VII. CONCLUSIONS

In this paper the remote control for a Lego Mindstorms mobile robot has been presented. The remote experiment has been developed inside the Automatic Control Telelab environment, a laboratory which allows the remote control of physical processes.

Fig. 10. The *Experiment Interface*.

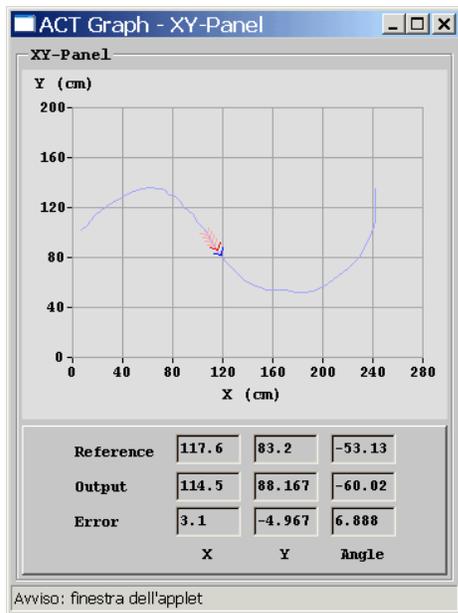Fig. 12. A Matlab X-Y plot for a previously performed experiment.

Fig. 11. The X-Y plot of the on-line experiment.

By means of this facility, students can perform robotics and automation experiments through the Internet in a safe and easy way. Users can design their controllers through a Simulink model and test it on the actual robot. Students who experienced this facility provided in general very positive and enthusiastic comments.

Future developments regard the student competition also for the Lego robot, and the possibility to create virtual obstacles which must be avoided by the robot. This last feature will allow students to design and test path planning algorithms.

REFERENCES

[1] A. Bicchi, A. Coppelli, F. Quarto, L. Rizzo, F. Turchi, and A. Balestrino, "Breaking the lab's walls: Tele-laboratories at the university of pisa," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, 2001, pp. 1903–1908.
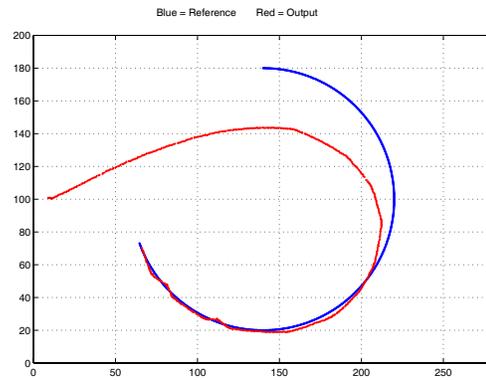
[2] P. Saucy and F. Mondada, "Khepontheweb: open access to a mobile robot on the internet," *IEEE Robotics and Automation Magazine*, vol. 7, no. 1, pp. 41–47, March 2000.

[3] S. Poindexter and B. Heck, "Using the web in your courses: What can you do? what should you do?" *IEEE Control Systems Magazine*, pp. 83–92, Febraury 1999.

[4] S. Dormido, "Control learning: present and future," in 15*th IFAC World Congress b'02*, Barcelona, July 2002.

[5] C. Schmid, "The virtual lab VCLAB for education on the web," in *Proc. of IEEE American Control Conference*, Philadelphia, June 1998, pp. 1314–1318, www.esr.ruhr-uni-bochum.de/VCLab/.

[6] B. Aktan, C. Bohus, A. Crowl, and M. Shor, "Distance learning applied to control engineering laboratories," *IEEE Transactions on education*, vol. 39, no. 3, pp. 320–326, August 1996.

[7] V. Ramakrishnan, Y. Zhuang, S. Hu, J. Chen, C. Ko, B. M. Chen, and K. Tan, "Development of a web-based control experiment for a coupled tank apparatus," in *Proc. of IEEE American Control Conference*, Chicago, June 2000, pp. 4409–4413, vlab.ee.nus.edu.sg/vlab/control/.

[8] T. Junge and C. Schmid, "Web-based remote experimentation using a laboratory-scale optical tracker," in *Proc. of IEEE American Control Conference*, Chicago, June 2000, pp. 2951–2954.

[9] J. Apkarian and A. Dawes, "Interactive control education with virtual presence on the web," in *Proc. of IEEE American Control Conference*, Chicago, June 2000, pp. 3985–3990, www.controlab.com.

[10] H. Hahn and M. Spong, "Remote laboratories for control education," in *Proc. of 39th IEEE Conference on Decision and Control*, Sydney, Australia, December 2000.

[11] G. Choy, D. Parker, J. d'Amour, and J. Spencer, "Remote experimentation: a web-operable two phase flow experimnet," in *Proc. of IEEE American Control Conference*, Chicago, June 2000, pp. 2939–2943.

[12] M. Casini, D. Prattichizzo, and A. Vicino, "The Automatic Control Telelab: a remote laboratory of automatic control," in *Proc. of 40th IEEE Conference on Decision and Control*, Orlando, Florida, December 2001.

[13] ——, "The Automatic Control Telelab: a user-friendly interface for distance learning," *IEEE Transactions on Education*, vol. 46, no. 2, pp. 252–257, 2003.

[14] T. L. Group, "The Lego mindstorms home page," 2003, [Online]. http://mindstorms.lego.com.

[15] J. Weinber and Y. Xudong, "Robotics in education: Low-cost platforms for teaching integrated systems," *IEEE Robotics and Automation Magazine*, vol. 10, no. 2, pp. 4–6, June 2003.

[16] O. Gerovich, R. Goldberg, and I. Donn, "From science projects to the engineering bench," *IEEE Robotics and Automation Magazine*, vol. 10, no. 3, pp. 9–12, September 2003.

[17] M. Casini, A. Garulli, D. Prattichizzo, and A. Vicino, "Remote system identification in the "Automatic Control Telelab" environment," in *Proc. of 42nd IEEE Conference on Decision and Control*, Maui, HI, December 2003, pp. 4956–4961.

[18] M. Casini, D. Prattichizzo, and A. Vicino, "E-learning by remote laboratories: a new tool for control education," in *Preprints of 6th IFAC Symposium on Advances in Control Education*, Oulu, Finland, June 2003, pp. 95–100.