

# Operating Remote Laboratories Through a Bootable Device

Marco Casini, *Member, IEEE*, Domenico Prattichizzo, *Member, IEEE*, and Antonio Vicino, *Fellow, IEEE*

**Abstract**—Remote laboratories are laboratories where the hardware devices are accessible through the Internet 24 h a day for running experiments on physical processes. These laboratories are commonly used as complementary tools in engineering education, allowing students to integrate theoretical notions with practical experiments without the necessity of being physically present inside a laboratory. The increased availability of experiments in a remote laboratory is certainly an advantage for students, but it implies an increase of the probability of hardware and software failures. Thus, maintaining physical processes in a remote laboratory may become a heavy task for laboratory administrators. Remote laboratory inefficiency due to hardware and software maintenance can be certainly reduced if the time that is required for software reinstallation and/or upgrade is kept as short as possible. In this paper, a technique based on the use of a bootable (live) device on the server side of a remote laboratory is reported. This solution, which, in principle, can be used in almost any remote laboratory, allows an administrator to greatly reduce the time that is needed to fix a hardware/software failure, as well as to add a new process to the laboratory in a faster way. Moreover, the use of such a tool will also increase the reliability of the entire laboratory. The application of the proposed method to the “Automatic Control Telelab,” which is a remote laboratory that was developed at the University of Siena, is also described in detail.

**Index Terms**—Bootable live device, distance learning, remote laboratories.

## I. INTRODUCTION

REMOTE laboratories represent an interesting emerging technology for distance learning in the engineering field [1]–[3]. By means of these laboratories, students can interact with physical processes 24 h a day from any location. In general, they only need a general-purpose personal computer (PC) with an Internet connection and a web browser to perform remote experiments.

Remote processes may be of various nature, ranging from mechanical to electronic, from hydraulic to thermal to chemical, etc., or a mixture of them (see, e.g., [4]–[7]). Several applications of remote laboratories also refer to the robotics field [8]–[12]. Depending on the requested tasks, students can set some variables and start the experiment. During the experiment, they can usually observe the signals of interest and a live video, whereas at the end of the experiment, they can download the measured data in order to perform offline analysis. Some remote laboratories also allow students to send user-specified

files in order to perform the requested task. This is, for instance, the case of remote laboratories of automatic control, where students can interact with physical systems by designing and testing controllers.

Remote laboratories are thus a powerful tool in engineering education, and they are usually designed to be easy to use, in order to reduce the time that is needed by students to understand how such facilities work. This allows students to concentrate their efforts toward the given task. To this purpose, remote laboratory architectures are usually based on well-known software environments, such as LabVIEW [13] or Matlab [14]. However, despite the ease of use from the user point of view, sometimes, maintaining such kind of laboratories requires great effort from the laboratory administrator. In fact, with the processes and the PCs being online 24 h a day, it is common that some hardware or software failures occur. Moreover, adding new processes or updating the laboratory with new software versions may be a time-consuming task.

The aim of this paper is to show how the use of a bootable (live) device—here, a CD—can improve both the reliability and the efficiency of remote laboratories. A live CD is a CD-ROM containing an operating system (OS) and other software, which can be run directly from the CD drive without any installation on a hard drive. Although the first live CD (Mac OS 7) is dated 1991, only in recent years has this technology considerably grown. At present, bootable devices are used in several applications, ranging, e.g., from web server maintenance [15] to network security [16] to forensic usage [17]. A list of several live CDs is available, for instance, in [18].

In this paper, the usage of a bootable CD in the remote laboratory context is described. In general, such a solution can be adopted in almost any remote laboratory, leading to an improvement of efficiency and reliability. The application of this technology to the “Automatic Control Telelab” (ACT), which is a remote laboratory that was developed at the University of Siena, is also described in detail. In addition to remote laboratories, it is the authors’ opinion that such a technology can be used with success in several engineering education frameworks.

This paper is organized as follows: Section II provides a short description of the ACT. In Section III, the proposed method and the main advantages, which drove us to develop it, are reported. In Section IV, a detailed description of the application to the ACT is reported, while in Section V, conclusions are drawn.

## II. ACT

The ACT is a remote laboratory that was developed at the University of Siena [19], [20]. The main goal of the laboratory

Manuscript received March 22, 2007; revised August 9, 2007.

The authors are with the Dipartimento di Ingegneria dell’Informazione, Università di Siena, 53100 Siena, Italy (e-mail: casini@ing.unisi.it; prattichizzo@ing.unisi.it; vicino@ing.unisi.it).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>

Digital Object Identifier 10.1109/TIE.2007.907026

is to allow students to put in practice their theoretical knowledge of control theory in an easy way and without restrictions due to laboratory opening time and process availability. In fact, the ACT is accessible 24 h a day from any computer that is connected to the Internet by means of any common Internet browser. To reflect its educational purpose, since 1999, the ACT has been used in control systems and robotics classes at the Engineering Faculty of University of Siena. Through this laboratory, students can run control and system identification experiments [21]. One more feature of the ACT is the so-called *student competition*, i.e., a mechanism through which students can compete to design the controller with the best performance for a given remote experiment [22].

The ACT is currently used by several institutions around the world, such as, e.g., the University of Pisa, the Polytechnic of Milan, and the Massachusetts Institute of Technology [23]. In addition to education, the ACT is also used for research purposes; an example of such applications is the control of a helicopter simulator with a controller based on the “Predictive Inverse Neurocontrol” [24] that was developed at the Far-Eastern State Technical University, Vladivostok, Russia.

A key feature of the ACT is that students can design their own controllers by means of the Matlab/Simulink environment [14], [25]. Since Matlab/Simulink is a standard tool in control systems classes, students do not need to learn additional tools to run experiments. They can simply design their controllers in an easy way by building a standard Simulink model that is similar to those that are commonly used to run simulations of dynamic systems and test them on a real remote experiment.

Running an experiment in ACT is an interactive experience: During the control experiment, students can change controller parameters and reference signals, and observe the signals of interest and a live video of the process. At present, five processes are available for remote experiments: a dc motor, a water tank, a magnetic levitation system, a two-degrees-of-freedom helicopter, and a Lego mobile robot (Fig. 1). It is worthwhile to remark that, in addition to the reported processes, which are mainly of mechanical/hydraulic nature, it is possible to connect to the ACT any other process that interfaces through a data acquisition board (DAQ), such as, e.g., thermal, electronic, and chemical processes. In the near future, an inverted pendulum will be added; through this system, a certain number of control laws could be used in order to swing up and stabilize the pendulum (see, e.g., [26]). The ACT home page is <http://act.dii.unisi.it>.

### III. LIVE CD ARCHITECTURE

A remote laboratory is conceptually divided into two parts: One is devoted to the connection and the control of the physical processes (server side), whereas the other is related to the communication with the user (client side). In this paper, the server side will be taken into account. The main issue in connecting physical processes to the remote laboratory is equipping the PCs that are devoted to the processes with special hardware and software, such as, for instance, DAQs and their drivers. A detailed description of the software that is needed by the ACT server is reported in Section IV.

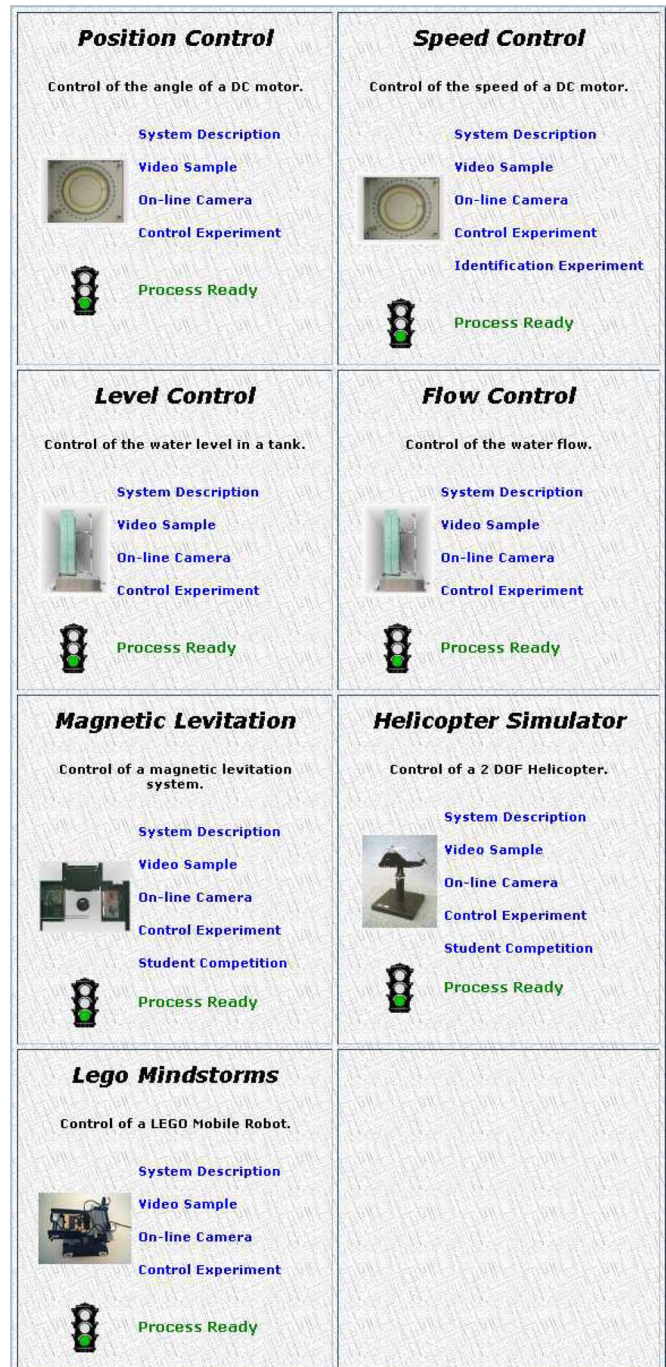


Fig. 1. ACT's remote experiments.

The proposed approach is to use a bootable (live) CD on the server side of a remote laboratory. A bootable CD is a CD-ROM that contains an OS along with other software, which can be run directly from the CD drive on system startup, without installing into permanent memory. Since the bootable CD does not require a hard disk to work, it can be used on PCs without hard disks, or if a hard disk is present, this CD does not alter the data that are stored in the device, unless specifically requested.

Using a bootable CD in a remote laboratory requires that all the software (and the OS) that are needed by the server must be stored on the CD. If a laboratory has many processes and experiments that are remotely available, it is likely that the software

that is installed on any server that is connected to a physical process is almost the same. This is specifically true if the architecture of the remote laboratory has been designed with the purpose of increasing the number of controlled systems with time.

By using a suitable server architecture, all the software that are common to any server can be stored in a unique live CD, while the files that are specific to a single process can be saved on a different device, such as a hard disk, a Universal Serial Bus (USB) storage unit, or a floppy disk. This allows one to use the same live CD for any server that is connected to any process. According to the approach that is proposed in this paper, the use of a hard disk to save process-specific files is not recommended. Moreover, the use of a floppy disk or of some kinds of USB pendrives allows one to mechanically set them as write protected to safeguard data against any sort of external attack. Of course, if data need to be permanently stored (such as, e.g., the data regarding a user registration), it is necessary to set the removable device to be writable. Alternatively, it is possible to store such data in a separate server. Note that, in this case, all the processes can share the same server.

It is worthwhile to note that, due to the increasing storage capabilities of USB pendrives, it is possible to avoid using the CD by storing all data in the pendrive. Although this solution has the advantage of requiring a single device, only the newest PCs are able to boot from a USB device. Moreover, by using a CD and a removable device, it is possible to split the software in two parts: 1) the software that is common to all processes (in the CD) and 2) the specific software for a given process (in the removable device). For these reasons, in this paper, the CD/pendrive configuration will be considered.

The main motivations for using a bootable CD are reported here.

- *Process update.* Adding a new process to a remote laboratory is a task that requires two main steps. The first is devoted to the physical process and regards hardware, i.e., connecting all hardware devices along with safety mechanisms. The second regards software installation and configuration. For the reasons that were previously explained, the time that is needed by the software installation can be greatly reduced by using a bootable CD (which contains all the needed software) and a configuration file that is stored in a floppy disk or USB pendrive.
- *Software update.* In addition to increasing the number of available processes, improvements on a remote laboratory can lead to the development of new functionalities, fixing of bugs, performance improvements, etc. In this case, one only needs to remaster the CD with the new software version and reboot the servers from CD. The updated version of the laboratory may then work on every process with minimal effort.
- *Failure restoring time.* In a remote laboratory, it is common that some software and/or hardware failures occur in some servers/processes. Regarding process failures (e.g., component breakdowns), it is, of course, impossible to find a standard way to repair them since they strongly depend on the specific process. However, other kinds of

hardware failures concerning the PC that is connected to the process can be avoided or easily and quickly fixed by using a bootable CD. For example, in the case of a hard disk failure, the process usually goes offline until a new hard disk is found, and all the software that is needed by the server is reinstalled and properly reconfigured. Since the number of software applications that are usually required by a remote laboratory is, in general, quite large (in Section IV, the list of software applications that are needed by the ACT is reported), this operation can take much time. Instead, by using a bootable CD that contains all the needed software, it is possible to connect processes to PCs with no hard disks, preventing such kinds of problems. Moreover, failures of other PC components can also be easily solved; in fact, it is only needed to temporarily replace the broken PC with any other (containing a DAQ) and to boot it from CD. Since the software in the CD does not need to read/write the hard disk, the original content of the hard disk is preserved. So, it is possible to use this PC without any problem until a new PC is ready to replace it.

In case of software failures, it is just needed to reboot the system to have the server fully operating. Moreover, the reboot procedure is not dangerous since the working data is written in memory [random access memory (RAM) disk] and not on permanent devices.

- *Increased reliability.* The reliability of a remote laboratory is increased by using a live CD for several reasons. The lack of a hard disk obviously prevents certain problems, such as, e.g., mechanical failures. Such kinds of failures, which can appear rare, may instead arise quite often, particularly, in view of the fact that such devices work continuously for 24 h a day for several years. One more reason for improved reliability is that the software installation is correct and all needed applications have been installed with the proper version. Finally, since the CD is a read-only storage device, it is free from virus corruption or other hacker attacks.

Notice that the CD does not need to continuously run 24 h a day. In fact, the CD loads the useful data to RAM only during the boot process, after which it stops. From this moment, all the applications run from memory, guaranteeing preservation of the CD drive as well as a fast execution time.

Once the proposed architecture has been adopted in the ACT, the previously described advantages have been confirmed in practice. In fact, for instance, the average time that a process is offline has been greatly reduced with respect to the previous architecture. Moreover, the time that is needed for a software upgrade has also become shorter.

#### IV. ACT SOFTWARE DESCRIPTION

Since any remote laboratory has its own features, it turns out to be difficult to describe the proposed approach in a detailed way without referring to a particular laboratory. To this purpose, in this section, the application to the ACT is reported in order

to describe a detailed list of software packages that are installed in the ACT bootable CD and in the removable device.

In the following, the live CD that is used in the ACT server will be denoted as “ACT-CD.” As reported in Section III, since the software inside the ACT-CD is independent from the remote process, it is necessary to have a device where the particular settings regarding any specific process must be stored. In what follows, a USB pendrive, which is referred to as “ACT-pen,” is considered.

The live CD that is chosen for the ACT is a remastered version of Knoppix [27], which is a bootable CD based on the Linux OS. The main features of Knoppix are the automatic hardware detection of a large number of peripherals, and the possibility to compress the data to store up to 2 GB on a CD-ROM or more than 8 GB on a DVD. Moreover, it is possible to add and remove applications to create a personalized bootable CD.

It is worthwhile to remark that, despite their diversities, all the ACT’s processes that were reported in Fig. 1 (except for the Lego mobile robot whose libraries work under a Microsoft Windows OS) run using copies of the same ACT-CD. Although the ACT-CD contains many software applications (present in the Knoppix CD by default), in the following, only the software that is strictly needed by the remote laboratory will be described. Note that several applications that are reported here are essential for almost any remote laboratory.

- *Linux OS.* As mentioned before, the use of a Knoppix-based CD allows one to have a live CD with a Linux OS inside. Since the ACT is able to work under Linux (in addition to Windows), it well fits to the Knoppix distribution.
- *Matlab/Simulink.* Since the ACT is based on the Matlab/Simulink environment, such a software must be stored in the CD. Of course, all the useful toolboxes, such as, for instance, the Real-Time Workshop (useful to convert a Simulink model to C code) [28], the Control System Toolbox, and the System Identification Toolbox, must also be stored in the CD.
- *GCC compiler.* Since the C code that was generated by the Real-Time Workshop needs to be compiled to obtain an executable, a C compiler is needed. In this case, the standard Linux GCC compiler has been included in the CD.
- *Apache web server with PHP.* Since the user has to connect directly to the server through a web browser, a web server must run on the server machine. So, the Apache HTTP server [29] has been included in the CD. Moreover, the PHP extension has also been installed since the web pages are generated by using this scripting language.
- *Comedi drivers.* Any remote laboratory needs to interface with a physical process in order to run control experiments. This task is usually done by using a DAQ, which allows for the analog-to-digital and digital-to-analog conversions of signals. Due to the high number of manufacturers and board models, it was chosen to install the Comedi drivers [30] into the CD. These drivers allow one to interface with more than 300 DAQs from the most

```
#!/bin/sh

#----- Network Parameters
IP_AD=192.168.112.49
IP_MK=255.255.0.0
GATEWAY=192.168.0.1
DNS=193.205.4.11
HOSTNAME=act-proc1.pendola.unisi.it
#-----
#----- DAQ Parameters
DAQ=ni_pcimio
#-----
#----- Camera Parameters
NUM_CAMS=1
#-----
#----- copy some files into ramdisk
cp -r . /ramdisk/home/knoppix
ln -s /home/knoppix/stat.txt /home/knoppix/www/stat.txt
#-----
#----- Load Comedi module
sudo modprobe $DAQ
sudo /usr/local/sbin/comedi_config /dev/comedi0 $DAQ
#-----
#----- Set webcam modules
sudo modprobe -r ov511
sudo modprobe ov511 cams=$NUM_CAMS lightfreq=50
#-----
#----- Start camera servers
webcam /home/knoppix/www/camera/act-camera.conf &
#-----
#----- Start SSH server
sudo cp shadow /etc/shadow
sudo chmod 700 /var/run/ssh
sudo /etc/init.d/ssh start
#-----
#----- Network Configuration
echo "nameserver $DNS" > /etc/resolv.conf
hostname $HOSTNAME
ifconfig eth0 $IP_AD netmask $IP_MK broadcast $IP_AD up
route add default gw $GATEWAY
#-----
#----- Start Apache with the given configuration file
apache -f /home/knoppix/httpd.conf -c "ServerName $IP_AD"
#-----
```

Fig. 2. Example of a startup file stored in the ACT-pen.

spread manufacturers, by using special C libraries, which allow one to interface with them in a transparent way with respect to the physical acquisition board.

- *Webcam software.* To allow users to see a live video of the experiment, a special software must run on the server. Many applications can perform this task, so the choice is not critical. Both the *webcam* [31] and the *camE* [32] software are used in the ACT. Of course, to allow this software to work, the proper video drivers (*video4linux*) must also be installed.
- *SSH server.* To allow an administrator to remotely connect to the server, an SSH server has been included in the CD. This allows the administrator to remotely connect in a safe way to the server, in order to perform some operations,

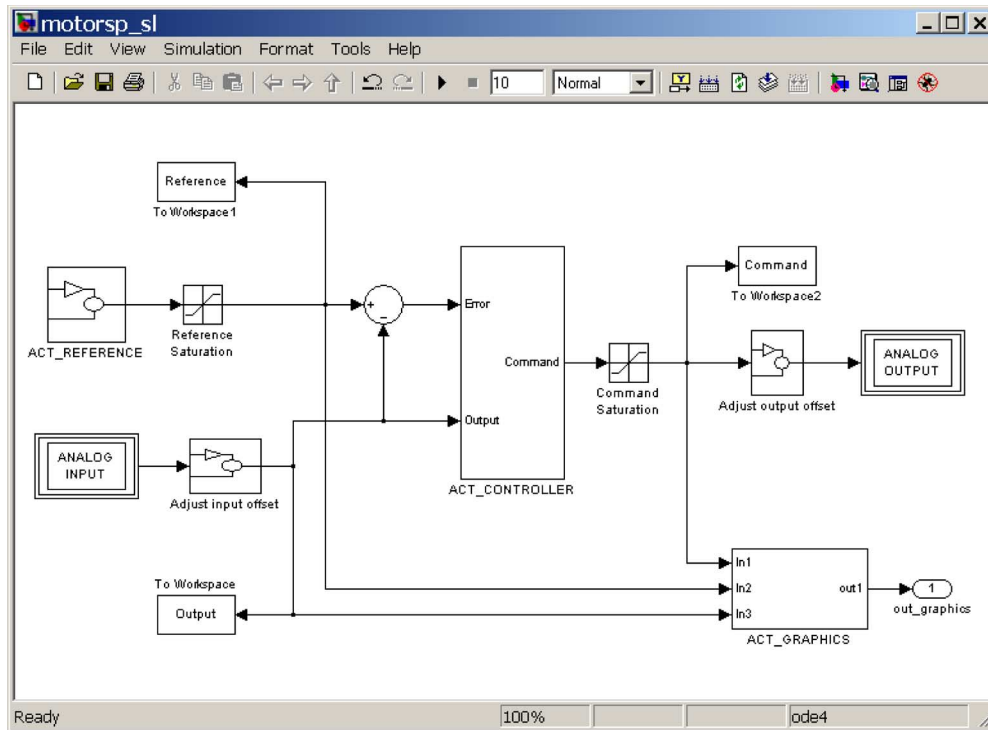


Fig. 3. Simulink model used for interfacing with a remote process.

such as, e.g., adding a new predefined controller for the process or updating files that describe the experiments.

- *ACT libraries.* Finally, all the libraries and the other files that were specifically written for the ACT have been included in the CD. Such libraries will be integrated in the C source code that was generated by Real-Time Workshop and allow for the interaction with the remote user as well as the real-time control of the process.

The main files that are stored in the ACT-pen are reported here.

- *Startup file.* The ACT-CD has been remastered, so that it automatically looks for a specific file in the removable device. Such a startup file is stored in the ACT-pen, and it is used to perform several initialization tasks, such as configuring the network parameters and the Comedi drivers, and starting the webcam software, the SSH server, and the Apache web server. An example of startup file is reported in Fig. 2.
- *Matlab license file.* A floating Matlab license file has been stored in the ACT-pen, in order to use Matlab. Although such a file could be included in the ACT-CD, putting this file in the ACT-pen allows one to easily change it once the license is renewed, without remastering the CD.
- *Simulink files.* For a given process, a Simulink file for each available experiment must be stored in the ACT-pen. For instance, two Simulink models must be provided for the dc motor since it is possible to perform both position and speed control experiments. Such models contain special blocks that interface with the DAQ as well as all the other blocks that are useful in defining the experiment behavior

and the storage of the input–output data. The Simulink model regarding the speed experiment on the dc motor is reported in Fig. 3. Since, in addition to user-defined controllers, students can also perform experiments by using predefined controllers, some Simulink models representing such predefined controllers may also be stored here, along with their compiled versions.

- *Description files.* For each experiment, a description file in .pdf format is stored in the ACT-pen, in order to allow students to have a detailed description of the process to be controlled. Moreover, a picture of the process can also be provided.
- *Web pages.* Since a user connects to the server through a browser, web pages must be provided by the server. These pages are stored in the ACT-pen, along with the Java applets, which provide the client interface. Note that almost all these files are the same for any experiment and do not need to be changed; only a few files containing specific data regarding an experiment, i.e., the name and number of inputs and outputs of the system, need to be manually defined. So, if desired, it is possible to store all these files in the ACT-CD.

A sketch of the overall architecture of the ACT is reported in Fig. 4. The gray boxes denote the applications that are stored in the ACT-CD.

## V. CONCLUSION

In this paper, the advantages of using a bootable CD on the server side of a remote laboratory have been presented. It has been shown how such a facility can increase both the reliability

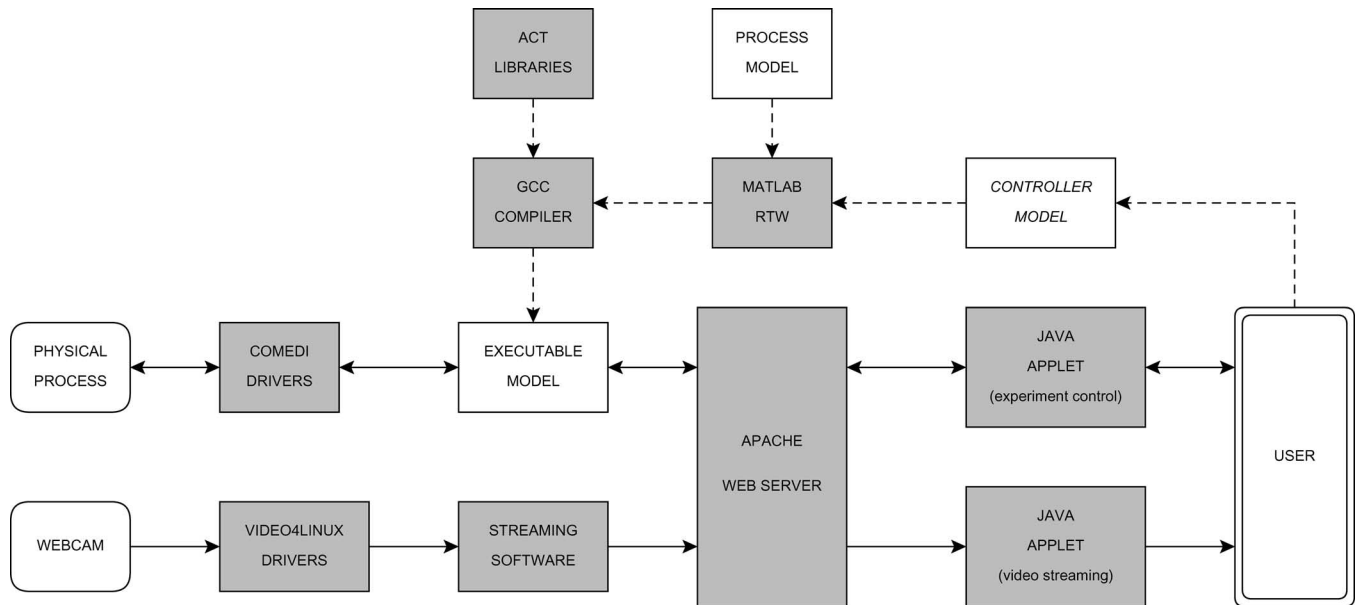


Fig. 4. Software architecture of the ACT. Since the software in gray background is independent of the physical process, it can be stored in a bootable CD.

and the efficiency of a remote laboratory. Since a live CD, which contains an OS (e.g., Linux) and other useful software, is a relatively new tool, it is the authors' opinion that such a tool can be used with success also in other frameworks of engineering education.

The design of the ACT has been updated to use a bootable CD on the server side, and the expected advantages have been confirmed in practice. In fact, the use of a Knoppix-based CD has increased the reliability of the whole system and has greatly reduced the time that is needed to add new processes to the remote laboratory, allowing laboratory administrators to easily maintain and increase the number of available processes for remote control. Although such changes are not directly visible to users, they are essential in providing a more reliable system whereby practical experiments can be performed.

## REFERENCES

- [1] S. Poindexter and B. Heck, "Using the web in your courses: What can you do? What should you do?" *IEEE Control Syst. Mag.*, vol. 19, no. 1, pp. 83–92, Feb. 1999.
- [2] H. Hahn and M. Spong, "Remote laboratories for control education," in *Proc. 39th IEEE Conf. Decision Control*, Sydney, Australia, Dec. 2000, pp. 895–900.
- [3] S. Dormido, "Control learning: Present and future," in *Proc. 15th IFAC World Congr. B*, Barcelona, Spain, Jul. 2002.
- [4] J. Overstreet and A. Tzes, "An Internet-based real-time control engineering laboratory," *IEEE Control Syst. Mag.*, vol. 19, no. 5, pp. 19–34, Oct. 1999.
- [5] G. Choy, D. Parker, J. d'Amour, and J. Spencer, "Remote experimentation: A web-operable two phase flow experiment," in *Proc. IEEE Amer. Control Conf.*, Chicago, IL, Jun. 2000, pp. 2939–2943.
- [6] 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. IEEE Amer. Control Conf.*, Chicago, IL, Jun. 2000, pp. 4409–4413.
- [7] R. Šafarič, M. Debevc, R. Parkin, and S. Uran, "Telerobotics experiments via Internet," *IEEE Trans. Ind. Electron.*, vol. 48, no. 2, pp. 424–431, Apr. 2001.
- [8] P. Saucy and F. Mondada, "Kheponthweb: Open access to a mobile robot on the Internet," *IEEE Robot. Autom. Mag.*, vol. 7, no. 1, pp. 41–47, Mar. 2000.
- [9] J. Kofman, W. Xianghai, T. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *IEEE Trans. Ind. Electron.*, vol. 52, no. 5, pp. 1206–1219, Oct. 2005.
- [10] R. Marín, 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. Ind. Electron.*, vol. 52, no. 6, pp. 1506–1520, Dec. 2005.
- [11] T. Chang, P. Jaroonsiriphan, M. Bernhardt, and P. Ludden, "Web-based command shaping of cobra 600 robot with a swinging load," *IEEE Trans. Ind. Informat.*, vol. 2, no. 1, pp. 59–69, Feb. 2006.
- [12] K. B. Sim, K. S. Byun, and F. Harashima, "Internet-based teleoperation of an intelligent robot with optimal two-layer fuzzy controller," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1362–1372, Jun. 2006.
- [13] *Getting Started With LabVIEW*, Nat. Instrum. Corp., Austin, TX, 2005.
- [14] *Getting Started With Matlab*, MathWorks, Natick, MA, 2006.
- [15] F. Cohen, "Bootable CDs," *New Secur.*, vol. 2001, no. 8, pp. 17–19, 2001.
- [16] G. Chamales, "The Honeywall CD-ROM," *IEEE Security Privacy Mag.*, vol. 2, no. 2, pp. 77–79, Mar./Apr. 2004.
- [17] E-fense, *The Helix Live CD Page*. [Online]. Available: <http://www.e-fense.com/helix/>
- [18] N. Brand, *Frozentech's LiveCD List*. [Online]. Available: <http://www.livecdlist.com/>
- [19] M. Casini, D. Prattichizzo, and A. Vicino, "The automatic control telelab: A user-friendly interface for distance learning," *IEEE Trans. Educ.*, vol. 46, no. 2, pp. 252–257, May 2003.
- [20] M. Casini, D. Prattichizzo, and A. Vicino, "The automatic control telelab: A web-based technology for distance learning," *IEEE Control Syst. Mag.*, vol. 24, no. 3, pp. 36–44, Jun. 2004.
- [21] M. Casini, A. Garulli, D. Prattichizzo, and A. Vicino, "Remote system identification in the 'Automatic Control Telelab' environment," in *Proc. 42nd IEEE Conf. Decision and Control*, Maui, HI, Dec. 2003, pp. 4956–4961.
- [22] M. Casini, D. Prattichizzo, and A. Vicino, "E-learning by remote laboratories: A new tool for control education," in *Proc. Preprints 6th IFAC Symp. Advances Control Edu.*, Oulu, Finland, Jun. 2003, pp. 95–100.
- [23] M. Ishutkina, E. Feron, M. Casini, and A. Vicino, "An Internet based laboratory for control of a safety critical system," in *Proc. IEEE Int. Conf. Syst., Man, and Cybern.*, Hague, Netherlands, Oct. 2004, vol. 3, pp. 2707–2712.
- [24] K. V. Zmeu, B. S. Notkin, and V. V. Stepaniuk, "Predictive inverse neuro-control," *Pac. Sci. Rev.*, vol. 6, no. 1, pp. 22–25, 2004.
- [25] *Getting Started With Simulink*, MathWorks, Natick, MA, 2006.
- [26] N. Muşkinja and B. Tovornik, "Swinging up and stabilization of a real inverted pendulum," *IEEE Trans. Ind. Electron.*, vol. 53, no. 2, pp. 631–639, Apr. 2006.
- [27] K. Knopper, *Knoppix Linux Live CD*. [Online]. Available: <http://www.knoppix.org>
- [28] *Getting Started With Real-Time Workshop*, MathWorks, Natick, MA, 2006.

- [29] T.A.S. Foundation, *Apache HTTP Server Project*. [Online]. Available: <http://httpd.apache.org/>
- [30] D. SchleeF, F. Hess, and H. Bruyninckx, *Comedi, Linux Control and Measurement Device Interface*. [Online]. Available: <http://www.comedi.org>
- [31] xawtv Homepage. [Online]. Available: <http://linux.bytesex.org/xawtv/>
- [32] LinuxBRIT, *camE*. [Online]. Available: <http://linuxbrit.co.uk/camE/>



**Marco Casini** (S'00–M'00) was born in Siena, Italy, in 1973. He received the Laurea degree in information engineering and the Ph.D. degree in control systems from the Università di Siena, Siena, Italy, in 1999 and 2003, respectively.

In summer of 2001, he was a Visiting Researcher with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge. In 2005, he joined the Dipartimento di Ingegneria dell'Informazione, Università di Siena, where he is currently an Assistant Professor. He is

the author of several publications in international journals and conference proceedings. His research interests include set-membership system identification and modeling, remote laboratories of automatic control, and modeling of environmental systems.



**Domenico Prattichizzo** (M'95) received the M.S. degree in electronics engineering and the Ph.D. degree in robotics and automation from the University of Pisa, Pisa, Italy, in 1991 and 1995, respectively.

In 1994, he was a Visiting Scientist with the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge. Since 2002, he has been an Associate Professor of robotics and automation with the Dipartimento di Ingegneria dell'Informazione, Università di Siena, Siena, Italy.

He is a coauthor of more than 150 papers in the area of robotics and automation. His research interests include distance learning, visual servoing, robotic grasping, haptics, and geometric control.

Prof. Prattichizzo was a Co-chair of the Second IEEE International Workshop on Control Problems in Robotics and Automation (Las Vegas, December 2002) and Program Co-chair of the first World Haptics Conference (Pisa, 2005). He has served as a member of the Editorial Board of the IEEE TRANSACTIONS ON ROBOTICS (2003–2007) and is currently an Associate Editor of the IEEE TRANSACTIONS ON CONTROL AND SYSTEMS TECHNOLOGIES and of the *Journal of Dynamics of Continuous, Discrete and Impulsive Systems (DCDIS) Series B: Application and Algorithms*. He is a Coeditor of the books *Control Problems in Robotics* (Springer-Verlag, 2003) and *Multi-Point Physical Interaction With Real and Virtual Objects* (Springer-Verlag, 2005).



**Antonio Vicino** (M'90–SM'96–F'00) was born in 1954. He received the Laurea degree in electrical engineering from Politecnico di Torino, Torino, Italy, in 1978.

From 1979 to 1982, he held several fellowships at the Dipartimento di Automatica e Informatica, Politecnico di Torino. From 1983 to 1987, he was an Assistant Professor of automatic control in the same department. From 1987 to 1990, he was an Associate Professor of control systems at the Università di Firenze, Firenze, Italy. In 1990, he joined

the Dipartimento di Ingegneria Elettrica, Università di L'Aquila, as a Professor of control systems. Since 1993, he has been with the Università di Siena, Siena, Italy, where he founded the Dipartimento di Ingegneria dell'Informazione and covered the position of Head of the Department from 1996 to 1999. From 1999 to 2005, he was the Dean of the Engineering Faculty. In 2000, he founded the Center for Complex Systems Studies, Università di Siena, where he is currently the Director. He is the author of more than 200 technical publications, a coeditor of two books on robustness in identification and control, and a Guest Editor of the Special Issue "Robustness in Identification and Control" of the *International Journal of Robust and Nonlinear Control* and of the Special Issue "System Identification" of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He has worked on the stability analysis of nonlinear systems and time series analysis and prediction. His current research interests include robust control of uncertain systems, robust identification and filtering, mobile robotics, and applied system modeling.

Dr. Vicino served as an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 1992 to 1996. He currently serves as an Associate Editor for *Automatica* and an Associate Editor at Large for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL.