Operating remote labs through a bootable device

Marco Casini, Domenico Prattichizzo, Antonio Vicino

Dipartimento di Ingegneria dell'Informazione
Università di Siena, Siena - Italy
Email: {casini,prattichizzo,vicino}@ing.unisi.it

***Abstract*— Remote labs are laboratories where the hardware devices are accessible through the Internet 24 hours a day for running experiments on physical processes. These labs are commonly used as complementary tools in engineering education, allowing students to integrate theoretical notions with practical experimentations without the necessity to be physically present inside a laboratory.**

The increased availability of experiments in a remote lab 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 lab may become a heavy task for lab administrators. Remote lab inefficiency due to hardware and software maintenance can be certainly reduced if the time 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 lab is reported. This solution, which in principle can be used in almost any remote lab, allows an administrator to greatly reduce the time needed to fix a hardware/software failure, as well as to add a new process to the lab in a faster way. Moreover, the use of such a tool will also increase the reliability of the entire lab. The application of the proposed method to the "Automatic Control Telelab", a remote lab developed at the University of Siena, is also described in details.

***Index Terms*— Remote labs, distance learning, bootable live device.**

I. Introduction

Remote laboratories represent an interesting emerging technology for distance learning in the engineering field [1], [2], [3]. By means of these labs students can interact with physical processes 24 hours a day from any location. In general, they only need a general purpose 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], [5], [6], [7]. Several applications of remote labs refer also to the robotics field [8], [9], [10], [11], [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 labs allow also students to send user-specified files in order to perform the requested task. This is for instance the case of remote labs of automatic control, where students can interact with physical systems by designing and testing controllers.

Remote labs are thus a powerful tool in engineering education, and they are usually designed to be easy to use, in order to reduce the time needed by students to understand how such facilities work. This allows students to concentrate their efforts toward the given task. To this purpose, remote labs architectures are usually based on well-known software environments, like, e.g., LabVIEW [13] or Matlab [14]. However, despite the ease of use from the user point of view, sometimes, maintaining such kind of labs requires a great effort from the lab administrator. In fact, being the processes and the PCs online 24 hours a day, it is not uncommon that some hardware or software failures occur. Moreover, adding new processes or updating the lab 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 labs. A live CD is a CD-ROM containing an operating system 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 this technology has 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 remote labs context is described. In general, such a solution can be adopted in almost any remote lab, leading to an improvement of efficiency and reliability. The application of this technology to the "Automatic Control Telelab" (ACT), a remote lab developed at the University of Siena, is also described in details. In addition to remote labs, it is the authors' opinion that such a technology can be used with success in several engineering education frameworks.

The paper is organized as follows. Section II provides a short description of the Automatic Control Telelab. 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. The Automatic Control Telelab

The Automatic Control Telelab (ACT) is a remote laboratory developed at the University of Siena [19], [20]. The main goal of the lab 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 hours a day from any computer 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 lab, 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 best performance for a given remote experiment [22].

The ACT is currently used by several institutions around the world, like 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] developed at the Far-Eastern State Technical University (FESTU) of 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 similar to those 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, 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 2-DOF 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 possibile to connect to the ACT any other process which interfaces through a data acquisition board, like, e.g., thermal, electronic, and chemical processes. In the near future, an inverted pendulum system will be added; through this system a certain number of control laws should 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.

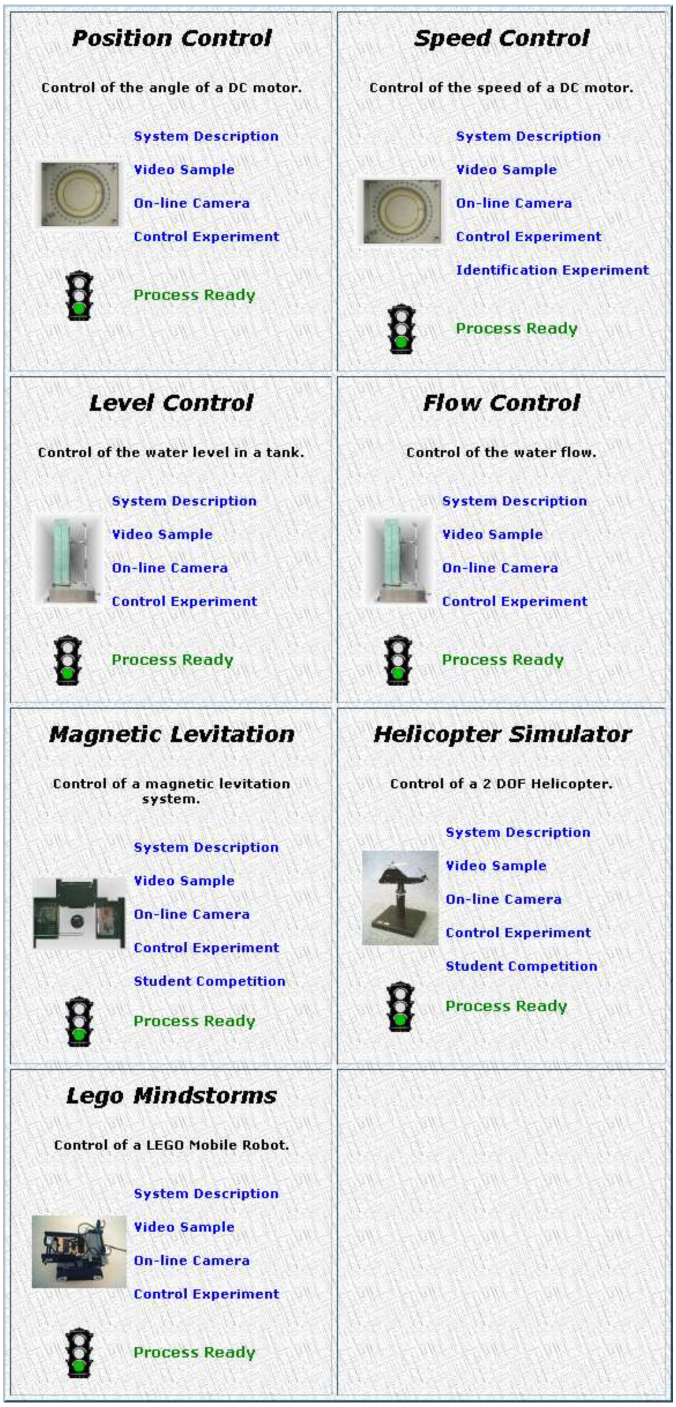

Fig. 1. The Automatic Control Telelab's remote experiments.

III. The live CD architecture

A remote lab is conceptually divided in 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 to connect physical processes to the remote lab is to equip the PCs devoted to the processes with special hardware and software, such as, for instance, data acquisition boards (DAQs) and their drivers. A detailed description of the software needed by the ACT server is reported in Section IV.

The proposed approach is to use a bootable (live) CD on the server side of a remote lab. A bootable CD is a CD-ROM which contains an operating system 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 stored in the device unless specifically requested.

Using a bootable CD in a remote lab requires that all the software (and the O.S.) needed by the server, must be stored on the CD. If a lab has many processes and experiments available remotely, it is likely that the software installed on any server connected to a physical process is almost the same. This is particularly true if the architecture of the remote lab has been designed with the purpose of increasing the number of controlled systems with time.

By using a suitable server architecture, all the software common to any server can be stored in a unique live CD, while the files specific to a single process can be saved on a different device like a hard disk, a USB storage unit or a floppy disk. This allows one to use the same live CD for any server connected to any process. According to the approach 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 to mechanically set them as write-protected to safeguard data against any sort of external attack. Of course, if data needs to be permanently stored (like, 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 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: the software common to all processes (in the CD) and the specific software for a given process (in the removable device). For these reasons, in this work, the CD/pendrive configuration will be considered.

In the following, the main motivations for using a bootable CD are reported.

- *Process update.* Adding a new process to a remote lab is a task which requires two main steps. The first is devoted to the physical process, and regards hardware: connecting all hardware devices along with safety mechanisms. The second regards software installation and configuration. For the reasons previously explained, the time needed by the software installation can be greatly reduced by using a bootable CD (which contains all the needed software) and a configuration file stored in a floppy disk or USB pendrive.
- *Software update.* In addition to increasing the number of available processes, improvements on a remote lab can regard the development of new functionalities, fixing bugs, performance improvements, etc. In this case, it is only sufficient to remaster the CD with the new software version and reboot the servers from CD. The updated version of the lab may then work on every process with a minimal effort.
- *Failures restoring time.* In a remote lab it is not uncommon 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 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 needed by the server is reinstalled and properly reconfigured. Since the number of software applications usually required from a remote lab is in general quite large (in Section IV, the list of software applications needed by the ACT is reported), this operation can take much time. Instead, by using a bootable CD which contains all the needed software, it is possible to connect processes to PCs with no hard disks, preventing from 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 data acquisition board) and booting 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 (ram-disk) and not on permanent devices.
- *Increased reliability.* The reliability of a remote lab is increased by using a live CD for several reasons. The lack of a hard disk obviously prevents from certain problems, like, e.g., mechanical failures. Such kinds of failures, which can appear rare, may instead arise quite often, especially in view of the fact that such devices work continuously 24 hours 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 viruses corruptions or other hacker attacks.

  Notice that the CD does not need to continuously run 24 hours 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, also the time needed for a software upgrade has become shorter.

IV. ACT SOFTWARE DESCRIPTION

Since any remote lab has its own features, it turns out difficult to describe the proposed approach in a detailed way without referring to a particular lab. To this purpose, in this section, the application to the Automatic Control Telelab is reported in order to describe a detailed list of software packages installed in the ACT bootable CD and in the removable device.

In the following, the live CD 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, referred to as "ACT-pen", is considered.

The live CD chosen for the ACT is a remastered version of Knoppix [27], a bootable CD based on the Linux operating system. 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 over 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 reported in Fig. 1 (except for the Lego mobile robot whose libraries work under a Microsoft Windows O.S.) run using copies of the same ACT-CD. Although the ACT-CD contains lots of software applications (present in the Knoppix CD by default), in the following, only the software strictly needed by the remote lab will be described. Note that several applications reported below are essential for almost any remote lab.

- *Linux Operating System.* As mentioned before, the use of a Knoppix-based CD allows one to have a live CD with a Linux O.S. 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, the System Identification Toolbox, etc., must also be stored in the CD.
- *GCC compiler.* Since the C code 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 have 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 been also installed, since the web pages are generated by using this scripting language.
- *Comedi drivers.* Any remote lab needs to interface with a physical process in order to run control experiments. This task is usually done by using a data acquisition board, which allows for the A/D and D/A 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 data acquisition boards from the most 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 these 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, like, e.g., adding a new predefined controller for the process or updating files which describe the experiments.
- *ACT libraries.* Finally, all the libraries and the other files specifically written for the ACT have been included in the CD. Such libraries will be integrated in the C source code generated by RTW and allow for the interaction with the remote user as well as the real-time control of the process.

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

- *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 which interface with the DAQ as well as all the other blocks useful to define the experiment behavior and the storage of the i/o 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

```
#!/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/sshd
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.

almost all these files are the same for any experiment and do not need to be changed; only 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 Automatic Control Telelab is reported in Fig. 4. The gray boxes denote the applications stored in the ACT-CD.

V. Conclusions

In this paper, the advantages of using a bootable CD on the server side of a remote lab have been presented. It has been shown how such a facility can increase both the reliability and the efficiency of a remote lab. Since a live CD which contains an operating system (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 Automatic Control Telelab 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 needed to add new processes to the remote lab, allowing lab 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 to provide 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 Systems Magazine*, pp. 83–92, Febraury 1999.

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

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

[4] J. Overstreet and A. Tzes, "An internet-based real-time control engineering laboratory," *IEEE Control Systems Magazine*, vol. 19, no. 5, pp. 19–34, October 1999.

[5] 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, IL, June 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. of IEEE American Control Conference*, Chicago, IL, June 2000, pp. 4409–4413.

[7] R. Šafarič, M. Debevc, R. Parkin, and S. Uran, "Telerobotics experiments via internet," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 2, pp. 424–431, 2001.

[8] 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.

[9] J. Kofman, W. Xianghai, T. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 5, pp. 1206–1219, 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 Transactions on Industrial Electronics*, vol. 52, no. 6, pp. 1506–1520, 2005.

[11] T. Chang, P. Jaroonsiriphan, M. Bernhardt, and P. Ludden, "Web-based command shaping of cobra 600 robot with a swinging load," *IEEE Transactions on Industrial Informatics*, vol. 2, no. 1, pp. 59–69, 2006.

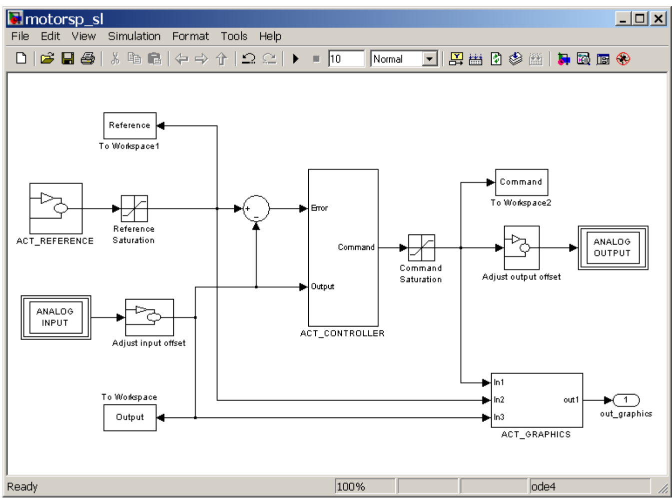

Fig. 3. Simulink model used for interfacing with a remote process.

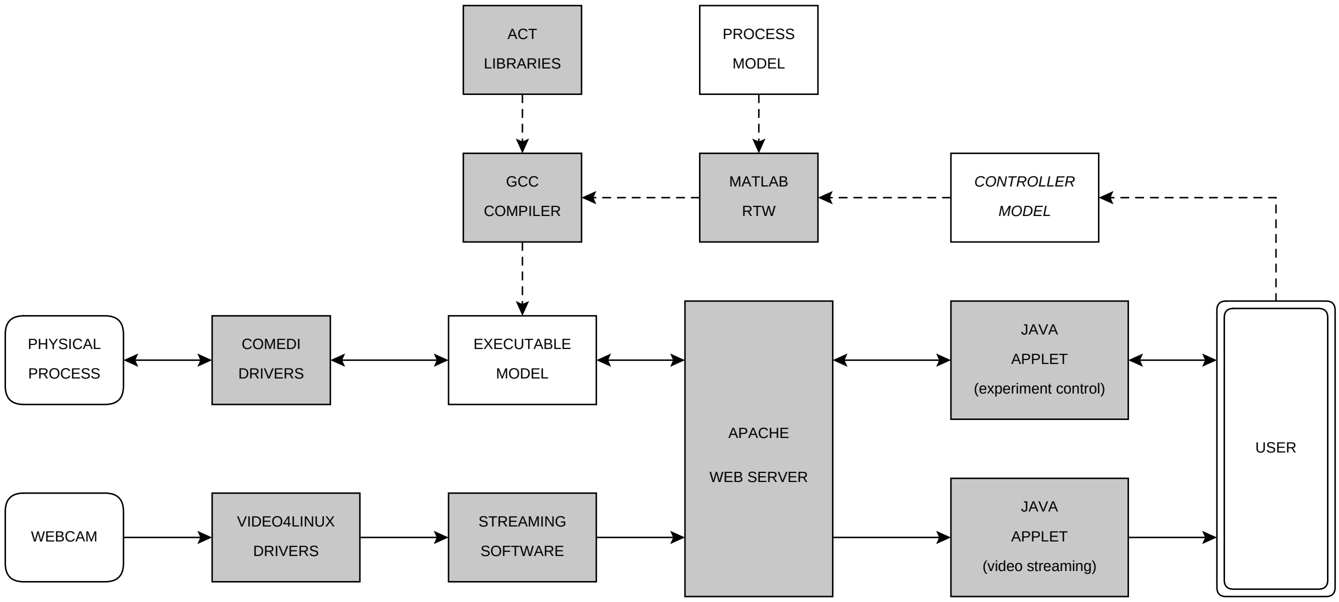

Fig. 4. Software architecture of the Automatic Control Telelab. Since the software in gray background is independent of the physical process, it can be stored in a bootable CD.

[12] K. B. Sim, K. S. Byun, and F. Harashima, "Internet-based teleoperation of an intelligent robot with optimal two-layer fuzzy controller," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 4, pp. 1362–1372, 2006.

[13] N. Instruments, *Getting started with LabVIEW*. Austin, TX: National Instruments Corporation, 2005.

[14] T. Mathworks, *Getting started with Matlab*. Natick, MA: The Mathworks, Inc., 2006.

[15] C. F., "Bootable CDs," *Network Security*, vol. 2001, no. 8, pp. 17–19, 2001.

[16] C. G., "The Honeywall cd-rom," *IEEE Security & Privacy Magazine*, vol. 2, no. 2, pp. 77–79, 2004.

[17] E-fense, "The Helix live cd page," [online], http://www.e-fense.com/helix/.

[18] B. N., "Frozentech's livecd list," [online], http://www.livecdlist.com/.

[19] M. Casini, D. Prattichizzo, and A. Vicino, "The Automatic Control Telelab: a user-friendly interface for distance learning," *IEEE Transactions on Education*, vol. 46, no. 2, pp. 252–257, 2003.

[20] ——, "The Automatic Control Telelab: A web-based technology for distance learning," *IEEE Control Systems Magazine*, vol. 24, no. 3, pp. 36–44, 2004.

[21] 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.

[22] 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.

[23] M. Ishutkina, E. Feron, M. Casini, and A. Vicino, "An internet based laboratory for control of a safety critical system," in *Proc. IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, The Hague, The Netherlands, October 2004, pp. 2707–2712.

[24] K. V. Zmeu, B. S. Notkin, and V. V. Stepaniuk, "Predictive inverse neurocontrol," *Pacific Science Review*, vol. 6, no. 1, pp. 22–25, 2004.

[25] T. Mathworks, *Getting started with Simulink*. Natick, MA: The Mathworks, Inc., 2006.

[26] N. Muşkinja and B. Tovornik, "Swinging up and stabilization of a real inverted pendulum," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 2, pp. 631–639, 2006.

[27] K. Knopper, "Knoppix Linux live cd," [online], http://www.knoppix.org.

[28] T. Mathworks, *Getting started with Real-Time Workshop*. Natick, MA: The Mathworks, Inc., 2006.

[29] T. A. S. Foundation, "Apache HTTP Server Project," [online], http://httpd.apache.org/.

[30] D. Schleef, F. Hess, and H. Bruyninckx, "Comedi, Linux control and measurement device interface," [online], http://www.comedi.org.

[31] K. G., "xawtv homepage," [online], http://linux.bytesex.org/xawtv/.

[32] LinuxBRIT, "camE," [online], http://linuxbrit.co.uk/camE/.