Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-29 (2015) 117–122

A LOW-COST OPEN SOURCE HARDWARE IN CONTROL EDUCATION. CASE STUDY:ARDUINO-FEEDBACK MS-150

P. Reguera, D. García, M. Domínguez, M.A. Prada, S. Alonso

Instituto de Automática y Fabricación. Área: Automática y Control.
Escuela de Ingenierías Industrial e Informática, Campus Universitario de Vegazana,
Universidad de León. 24071 León, España. Tfno: +34-987-291694.
e-mail: {prega, dgarcp01.estudiantes, manuel.dominguez, ma.prada, saloc}@unileon.es

Abstract: In this paper, we present a low-cost approach to upgrade an old Feedback system in order to widen the number and type of possible experiments in control education. This approach combines several different software technologies: C language for microcontroller programming, Matlab to chart data, Simulink to control the system, AJAX and XML for the client-side application, web services in the server-side application and serial I^2-C bus interface to read/write data from/to Feedback system. The ideas here explained could be applicable to any other physical system.

Keywords: **Control Education, Interconnection Technology, Remote Control, Laboratory Education, Arduino, Raspberry Pi**

1. INTRODUCTION

Nowadays, control education usually relies on the available simulation software packages and virtual laboratories. The most important problem in these virtual environments is that closed loop experiments are too often limited and students lack the physical feedback about the impact of control algorithms, their parameters, noise in signal acquisition or a fit sampling time (Ma et al., 2006; Fishwick, 1996).

Nevertheless, the field of control education deserves a special attention since several years ago, we can find technological structures designed to share remote devices through the Internet in order to enable hands-on sessions on real equipment located in various geographical locations. These structures are called remote or on-line laboratories (Aktan et al.,. 1996; Antsaklis et al, 1999; Domínguez et al., 2001; Dormido, 2002).

In that sense, the remote laboratory of automatic control at the University of León (LRA-ULE) has been used in control education providing remote access to physical systems. These systems generally have industrial and educational character (Prada et al.,2015; Fuertes et al., 2013; Fuertes et al., 2012).

However, the aim of this paper is to propose a cheap solution to connect modern or old physical systems to LRA-ULE. Connecting, in this context, involves the possibility of controlling and sharing old and new physical systems. Our assumption is that low-cost open source hardware can be used to provide that connectivity.

Therefore, in this paper, we present the upgrade of an old DC Motor (Feedback MS-150) with the final goal of enabling a low-cost connection with the LRA-ULE remote laboratory. This paper is structured as follows: section 2 shows our education targets; section 3 describes the physical system (Feedback) and microcontroller (Arduino) that are used in this experience; section 4 explains the hardware needed for Feedback control; section 5 shows data charting options; section 6 describes considerations about Arduino and remote network connection whereas section 7 explains the use of Raspberry Pi 2 instead of Arduino. Finally, section 8 discusses conclusions and future work.

2. EDUCATION TARGETS

A wide variety of controls experiments can be designed around a DC-motor unit. There are many commercial products as Feedback MS-150, with built-in electronics to power DC-motor. This is a good physical system but quite expensive. The equipment is available for our students to practice control-related topics in hands-on and remote sessions (Reguera et al., 2004).

Indeed, we have developed a set of control practices around the Feedback system: interactive simulations, hands-on sessions and remote connection (position PID simulation and parametrization, relay control, velocity control loop, etc.).

Nevertheless, it seems that bringing the Feedback system up-to-date by means of low-cost open source hardware would be very valuable to achieve the following targets:

1. Upgrade the Feedback system to expand the control practices that students can perform. For instance, providing a new set of DIY (Do It Yourself) control and system identification tutorials to the students.
2. Enable configurable practical tasks from a PC, such as changing the parameters of control algorithms, performing single or double loop controls, using any

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.11.223

kind of input signals, or changing the control algorithm running on microcontroller.

3. Hardware elements should be low-cost, so that students can develop their own physical control structure. The idea is that cost is not an obstacle to students learning, so that they can develop their own skills (applying the knowledge acquired to other problems) and complement their degree´s core competences.

4. The signal acquisition structure that is built must be easy/clear, so that students must understand concepts as sampling, jitter, signal filtering, aliasing and other similar concepts.

5. Make a compact, rugged, reliable and open structure to being integrated into LRA-ULE lab.

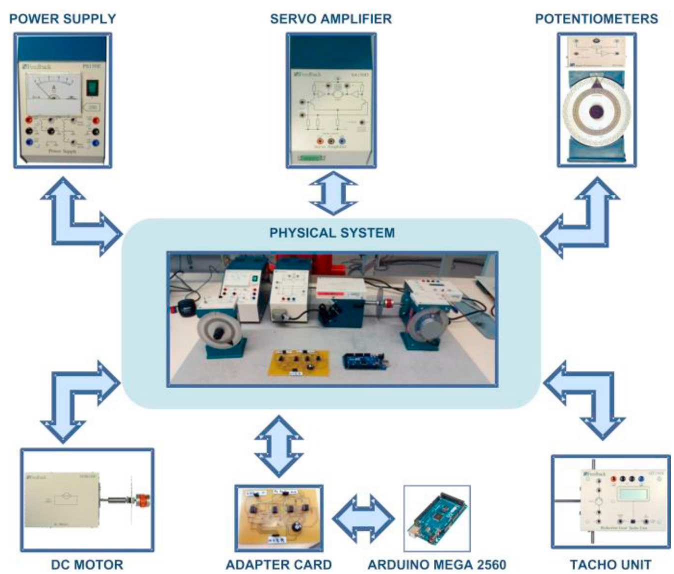

Fig. 1. Feedback MS-150

3. PHYSICAL SYSTEM AND LOW-COST HARDWARE

3.1 Physical system

The physical system is a Feedback MS-150 modular servo system marketed by Feedback Instruments. We have 4 equivalent units (the modules provide the same functionality but some are older than others) used by students in hands-on and remote sessions (Reguera et al., 2004, Fuertes et al., 2013; Fuertes et al., 2012). The Feedback MS-150 system used in this paper includes the next modules (see Fig. 1): a PS150E ±15 V Power Supply, a DCM150F DC Motor supplied with the SA150D Servo Amplifier SA150D, which allows changing motor rotation direction, an IP150H Input Potentiometer for manual setpoint, an OP150K Output Potentiometer to measure shaft rotation angle, and a GT150X Tachometer Unit to measure shaft rotation speed (Fuertes et al., 2013).

The upgrading of the Feedback MS-150 involves sampling and reading all signals of the system. This task allows system identification practices and the possibility of coding any control algorithm, not only a regular PID. For that purpose, a microcontroller and electronic devices are needed. An Ethernet device would also be needed to connect the system to the network.

3.2 Low-cost hardware selection

Nowadays, there are many types of low-cost open source microcontroller boards based on AVR or ARM processors with prices below 60 € (microcontroller + adapter card). This is a viable cost for students. Among these alternatives, Arduino Mega 2560 and Raspberry Pi 2 are two widely-used and documented microcontroller boards that are well known by students.

Table 1. Arduino Mega Vs Raspberry 2 Model B

Specs.	Arduino	Raspberry 2
Processor	ATmega2560 (AVR)	BCM2836 (ARM)
Proc. speed	16 MHz	900 MHz
Architecture	8 bit	32 bit
RAM mem.	256 KB	1 GB
Ethernet	✗	10/100 Base T
OS	✗	Linux RISC OS
Prog. lang.	C variant	Python, C,...any
Digital pins	54 (15 PWM)	40 GPIO
Analog pins	16 (0-5 V)	✗
USB	✗	4
Price	36 €	33 €

Considering Table 1, the appropriate choice would seem to be Raspberry (better specifications at the same price), but Arduino started in 2005 as a project for students and Raspberry is on the market just from 2012. Students have better knowledge about the Arduino philosophy and many of them own Arduino kits. Due to this, we have selected Arduino microcontroller.

4. CONTROL HARDWARE STRUCTURE

The initial control structure (with the Arduino microcontroller and an adapter card) addresses the first 4 mentioned educational targets: upgrading the system, enabling configurable practices and providing a low-cost solution with easy signal acquisition. This structure (see Fig. 2) reads 3 analog signals (shaft rotation angle, shaft rotation speed, manual setpoint) and writes 2 analog signals (servo amplifier input1 and input2). With this structure, the following practical tasks are possible:

- Testing a closed-loop position (or speed or both) control. The control algorithm could be PID, relay, time-optimal PID, or others.

- PID tuning, experiencing the windup effect and testing PID anti-windup.

- System identification experiments, using Strejc and Kupfmuller methods, ARX, ARMAX or OE models.

- Studying the effect of noise and disturbances in any part of the control loop.

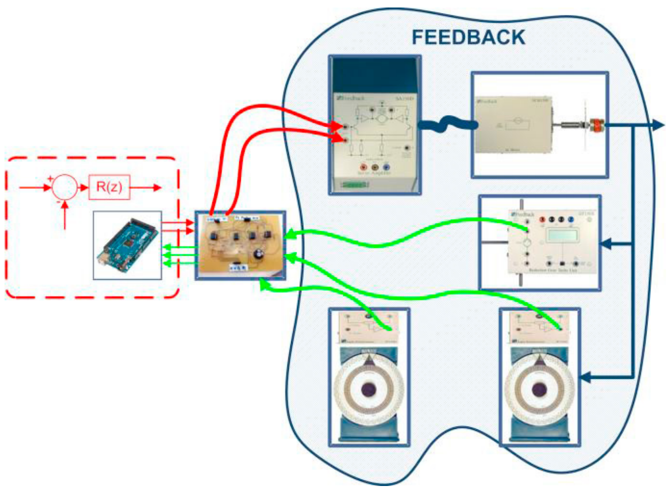

Fig. 2. Connections to the control structure

4.1 Adapter card

The main function of this card (see Fig. 3) is adapting voltages from the ±15V Feedback system to 0-5V Arduino´s I/O. We designed this adapter card so that a Raspberry Pi 2 microcontroller (3.3V GPIO) is also pluggable.

The control of the feedback system needs 2 analog outputs: input1 (right) and input2 (left) of the SA150D Servo Amplifier. The DC motor rotates to the right when input1>input2 and it rotates to the left when input2>input1.

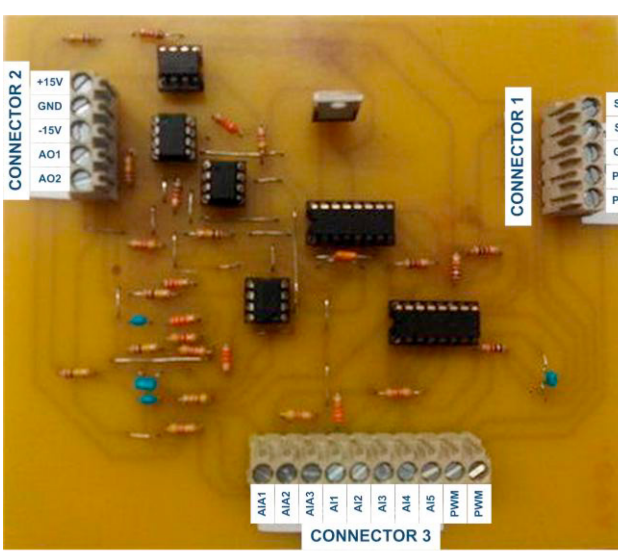

Fig. 3. Adapter card. Connector1: Arduino communication and PWM outputs. Connector2: supply and analog outputs. Connector3: analog inputs and PWM outputs.

Arduino has analog inputs, but not proper analog outputs. Typically, Arduino´s analog outputs are implemented generating 500 Hz PWM signals varying voltage between 0-5V. In this case, the problem with PWM for analog output generation is not control related: the frequencies of the PWM signals fall within the audible spectrum, resulting in a sharp annoying noise. For that reason, two PCF8591 integrated circuits have been used for the control output. The PCF8591 is a single-chip, single supply low-power 8-bit CMOS data acquisition device with 4 analog inputs, 1 analog output and a serial I^2C bus interface.

That is, no Feedback system signals are connected directly to Arduino, but through A/D and D/A converters. In addition, the serial I^2C bus interface enables the cascade connection of 8 similar chip devices. Therefore, it would possible to make an adapter card to read 32 analog inputs and write 8 analog outputs, if necessary.

Finally, the adapter card is also used to suppress high frequency noises for both input and output signals, because Arduino needs a very stable supply to avoid malfunction in the running programs. For that purpose, a RC passive low-pass filter with a cutoff frequency of 1 KHz (R=4.7 KΩ; C=33nF) was used.

4.2 Sampling considerations

The clock frequency of Arduino is 16 MHz, a low frequency, but enough to sample data from the Feedback system and control it. The sampling time mainly depends on the cycle time of the program running in Arduino. The minimum sampling time is over 10 ms with the control algorithm running in Arduino. If the Arduino board was used as data acquisition system (external mode), the minimum sampling time would be over 1 ms.

The A/D and D/A converters use serial I^2C bus interface. This communication is serial, so sampling time depends on number of chips used.

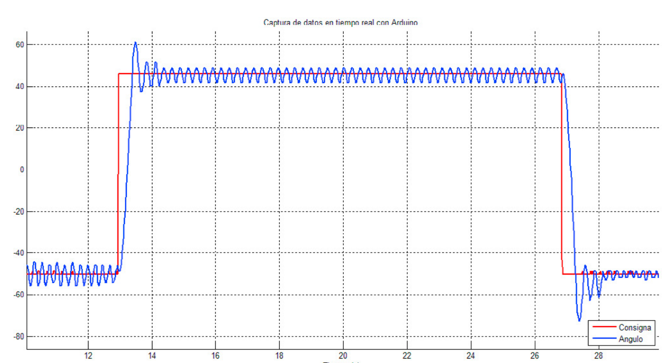

Fig. 4. Charting data with Arduino running control. Step response. Critical oscillation response used for Ziegler-Nichols PID tuning rule.

5. DATA CHARTING

From an educational point of view, charting signals from control loop is very important. It is only necessary, a serial connection from Arduino to Matlab to collect data. This software is used because our students know how to use it.

There are however 2 options for Arduino-Matlab connection: using only Matlab or using Matlab+Simulink.

5.1 Option 1. Matlab connection.

In this case, Arduino runs the control algorithm and the serial communication with Matlab can be used for monitoring (charting) and backing up the data for further analysis. (see Fig. 4)

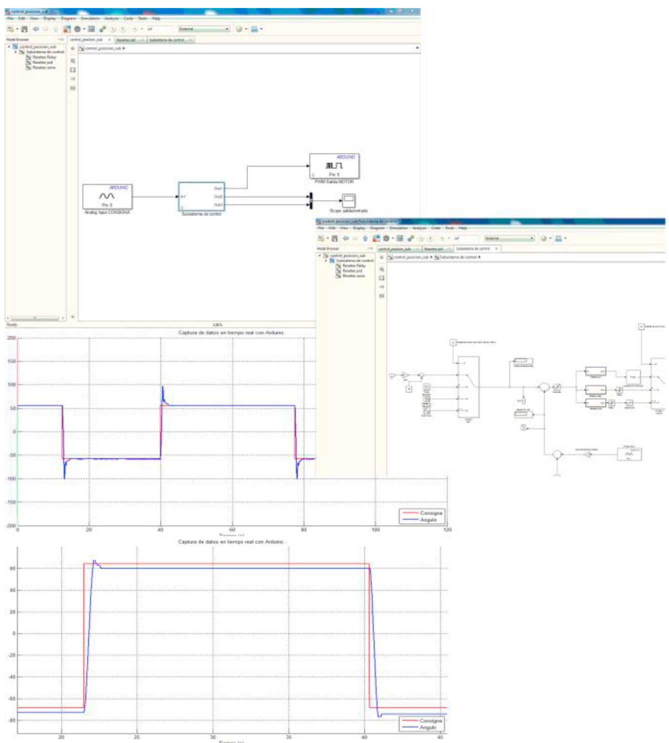

Fig. 5. Charting data with Arduino in external mode. Simulink runs control algorithm.

Students can follow a DIY tutorial to code a control program that runs in Arduino with the Arduino IDE. The program structure is common and students only need to change the algorithm. Students should have previous programming skills in C to be able to perform this task.

5.2 Option 2. Matlab connection + Simulink.

In this case, it is necessary to use Simulink (a block diagram environment for simulation, automatic code generation and continuous test and verification of embedded systems).

Table 2. Arduino Ethernet Shield

Specification	Arduino Ethernet Shield
Operating voltage	5V
Ethernet controller	W5100
Internal buffer	16 Kb
Connection speed	10/100 Mb
Arduino communication	SPI port
Ethernet connector	RJ45
Memory expansion	SD slot (class 6)
Price	30 €

Simulink has a library of blocks that connect to Arduino I/O, such as digital input and output, analog input and output (PWM), serial receive and transmit and servo read/write. (see Fig. 5). With this setup, there are two possibilities:

- Arduino runs the control algorithm and Simulink is used to implement the control, upload it to Arduino and run it. No programming skill is necessary.
- Arduino is used as an I/O card (external mode). Simulink is used to run control from the computer. Again, no programming skill is necessary.

6. REMOTE OPERATION

In previous sections, we have discussed how to upgrade the Feedback system to perform new control practices in hands-on sessions. But we would need an Ethernet device to connect this system to our remote laboratory.

Arduino needs new hardware, e.g., an Arduino Ethernet Shield expansion card that costs 30 € (Table 2). This price is very similar to the one of the Arduino Mega board itself. On the other hand, Raspberry Pi has an Ethernet port by default.

Unlike the solution adopted by (Kalúz et al., 2014), who focused on building a low-cost remote lab (the ArPi Lab), we try to optimize the physical system (increasing the number of possible practical sessions at the lowest cost). The final aim is connecting it to an already existing laboratory (the LRA-ULE lab) to read/write data and to control it. We have been implemented and tested 3 options (see Fig. 6):

- HTTP: Using Arduino as a web server.
- AJAX: Using Arduino as a web server and sending data to an AJAX client running on a web browser.
- Web socket: Arduino as a web server establishing full-duplex single-socket connection. The client side is a web browser.

6.1 HTTP Option

In this option, the Arduino Ethernet shield works as a web server. The Ethernet shield offers a storage space (SD card), where the HTML files are stored. Actually, only an "index.html" is necessary to view current and past Feedback system data (although it is possible creating as may files as needed). This web page shows historical chart data and enables submission of setpoint or control parameters (see Fig. 7).

When a browser visits that web page, an HTTP request is sent to the web server (Ethernet card) that hosts it. The web server acknowledges this request and sends back the response. In order to get the most up-to-date "real-time" information, the page could be constantly refreshed manually, but obviously that is not a great solution to chart data from control structure. Indeed, whenever a client wants to chart new data, the index.html is deleted and re-created in the SD card, whose usual lifetime is in the order of 100,000-1,000,000 write cycles. If that request would be submitted every second, the card might have a lifetime between 27-270 hours.

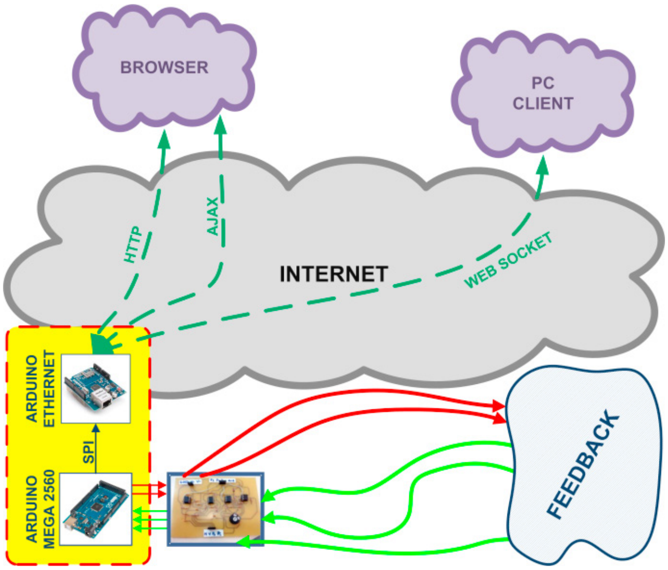

Fig. 6. Remote operation options using Arduino.

In short, it is not a good idea to use the Arduino that way to chart real-time data in this particular case. However, if the data rate were expected to be above 5 seconds, it could be a good and simple solution for process monitoring (remote supervision of industrial processes, home automation, etc.).

Fig. 7. Charting data in web browser.

6.2 AJAX Option

The idea is similar to the previous one, but we try to avoid continuous program/erase cycles of the index.html file in the SD card. The solution is AJAX. AJAX (Asynchronous JavaScript and XML) is a combination of techniques for updating parts of a web page, without the need of reloading the whole page in response to any request. It is widely used for creating fast and dynamic web pages (e.g., Google maps, Gmail, etc.).

For that purpose, a JavaScript code is included in index.html, so that Arduino generates an XMLHttpRequest object every time the web browser asks for new Feedback data. This idea improves the refresh rate of the charts rather than SD lifetime. Data can be represented as they come in the web browser and it is no necessary create the full index.html for each data or data set. When a user wants to write (to change setpoint, for example) on the Feedback system, a request is submitted to the Ethernet shield (and this time the index.html file is rewritten).

This is a good solution to let students visualize the behavior of the Feedback system in a simple way, without a third-party program such as Matlab, only with a web browser. Faculty can also use this solution in theoretical sessions as a tool to aid in the control education. However, it is not a robust solution to connect the system to LRA-ULE lab, because the Arduino Mega and the Arduino Shield are working very close to their physical limits (many tasks are running).

6.3 Websocket Arduino Option

HTTP WebSockets represent an evolution of web communications to provide a full-duplex, bidirectional communications channel that operates through a single socket over the Web. It provides a socket that is native to the browser, removing the overhead and dramatically reduces complexity. HTTP Web Sockets can provide up to a 1000:1 reduction in unnecessary HTTP header traffic and 3:1 reduction in latency.

This way, data visualization is faster and the "index.html" rewriting is largely avoided. But Arduino still works as a web server, which is not a robust solution. The main goal when we connect the Feedback system to LRA-ULE is to provide a web services architecture over Feedback system. But to achieve this, we would need a more powerful web server than the one we have tried in Arduino

Furthermore, although data can be acquired from the Feedback system with the proposed schema, problems would appear when the same schema is used with other physical systems that have more signals. The Ethernet Shield makes a great computational effort and the microprocessor would not be fast enough to work properly.

7. RASPBERRY Pi 2

The proposed Arduino schema allowed us to have an upgraded Feedback system that can be used for a variety of new practical tasks such as motor identification, filtering or velocity and position control with different algorithms. Now we can apply the technology adopted by (Kalúz et al., 2014), Raspberry Pi 2, to solve the remote communications problems that we find when we use Arduino. As mentioned above, the designed adapter card can be used with Raspberry Pi to read/write control signals from/to the physical system. The resulting new control schema is shown in (see Fig. 8).

At the time of writing this paper, we are testing the performance of this schema. The initial results are very promising. The idea is to take advantage of the capabilities of Raspberry Pi 2 (which is actually a little PC) and convert the Feedback system into an open connectivity system. The connection to LRA-ULE or any other remote laboratory would be very easy.

Nowadays, open connectivity is usually solved by means of web services, that structure their interface and operation in terms of the services provided to other programs or individuals. They use standardized protocols (TCP/IP and HTTP) and data formats (XML/JSON), which allow otherwise incompatible programs and platforms to communicate with one another.

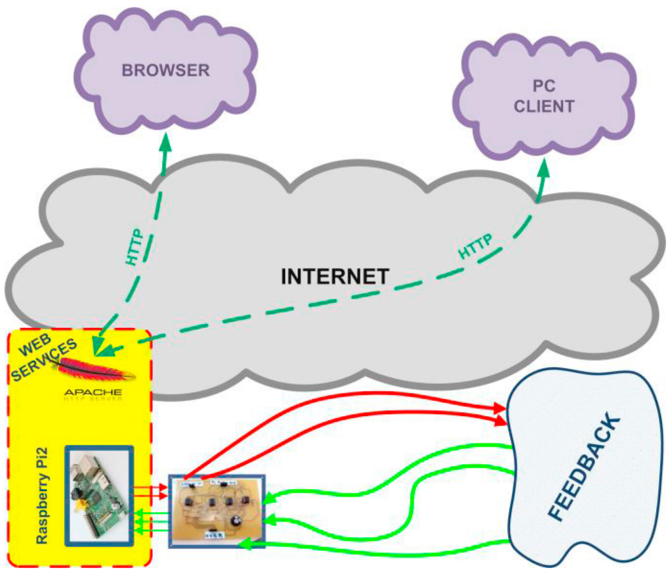

Fig. 8. Structure of the system with Raspberry Pi 2.

In this case, the Raspberry is invoked over network to process some requests or simply provide data from the Feedback system. The client can be a web browser or a compiled program. The ability of interoperating with different clients, regardless of the programming language in which they were coded, and the scalability provided by the clear separation between the system and the client, are the main advantages.

We are currently developing a set of web-services ("wspi2control ", "wspi2ident ", ...), which are written in PHP and use SOAP (Simple Object Access Protocol, one of the two choices in web services along with REST). They use the NuSOAP library that allows the development of a SOAP server in PHP, as well as the WiringPi library. Each web service exposes some methods to the clients: setGPIO(num,state), getGPIO(num,state), setSetPoint(value), etc.

8. CONCLUSIONS AND FUTURE WORK

We have upgraded the Feedback system using the proposed schema based on Arduino, so that a variety of tasks can be performed with the system: e.g., motor identification, signal filtering or velocity and position control based on the feedback value of the encoders. The proposed schema is a low-cost solution oriented to control education. The educational concepts applicable to the schema, as well as Arduino programming, are an attractive and intuitive learning way for students.

The adapter card used in this paper is an open card that allows reading/writing real analog signals. Students can develop their own, because they have the layout and the list of materials needed for the task.

In short, using Arduino, we have achieved the initial targets. However, the solution is not rugged and efficient enough and there are important connectivity problems that could be solved with a more powerful microcontroller board such as Raspberry Pi 2.

Finally, we propose the use of web services for remote operation. The Feedback system will provides a set of web services that run on demand. This means that interconnection with any structure is very simple and we can add new services or update existing web services without major changes to the client part of the a remote lab.

REFERENCES

Aktan, B., Bohus, C.A., Crowl, L.A., and Shor, M.H. Distance learning applied to control engineering laboratories. *IEEE transactions on Education*, 39(3), 320-326. 1996.

Antsaklis, P., Basar, T., De Carlo, R. (1999). *IEEE Control Systems Magazine*. Report on the NSF/CSS *Workshop on New Directions in Control Engineering Education*. Nº 19.

Domínguez, M., Marcos, D., Reguera, P., González, J.J., Blázquez, L.F., Connection Pilot Plant to the Internet, *IFAC Internet Based Control Education.* IBCE01, Madrid. Spain. 2001

Fishwick, P.A. ‘Web-based simulation: Some personal observations’, *in Proc. Winter Simulation Conference*, San Diego, 1996, pp. 772–779.

Fuertes, J.J., Domínguez, M., Prada, M.A., Alonso S., Morán, A., A virtual laboratory of D.C. motors for learning control theory, *International Journal of Electrical Engineering Education*, Manchester University Press, volume 50, 2013.

Fuertes, J.J., Alonso S., Morán, A., Prada, M.A., García, S., Del Canto, C.J., Virtual and Remote Laboratory of a DC Motor, *Advances in Control Education, volume 9*, 2012.

Kalúz, M., Cirka, L, Valo, R. and Fikar, M. ArPi Lab: A Low-cost Remote Laboratory for Control Education. *19th IFAC World Congress*. Cape Town, South Africa, August 24-29 2014.

Ma, J. and Nickerson, J.V. ‘Hands-on, simulated and remote laboratories: a comparative literature review’, *ACM Surveys*, 38 (2006), 1–24.

Prada, M.A., Fuertes, J.J., Alonso S., García, S., Domínguez M., Challenges and solutions in remote laboratories. Application to a remote laboratory of an electro-pneumatic classification cell), *Computers & Education, Elsevier*, 2015.

Reguera, P., Fuertes, J.J., Domínguez, M., Operating systems resources for web-based training in engineering education. *IFAC Internet Based Control Education IBCE04*, Grenoble, France 2004.