USING LOW-COST OPEN SOURCE HARDWARE TO CONTROL PUMA560 MOTORS

P. Reguera, S. Alonso, M. Domínguez, M.A. Prada, A. Morán, J.J. Fuertes

SUPPRESS Research Group. University of León. 24071 León, Spain. Tfno: +34-987-291694.
e-mail: {prega, saloc, manuel.dominguez, ma.prada, a.moran, jjfuem}@unileon.es

---

Abstract: In this paper, we present a low-cost approach to upgrade an outdated PUMA (Programmable Universal Machine for Assembly) 560 robot in order to widen the number and type of possible hands-on experiments in control and instrumentation education. Another aim of the upgrade is to enable its future connection to a remote laboratory. We propose a scalable structure to control and monitoring PUMA560 motors. This approach combines several different technologies: A Raspberry Pi for control and monitoring, Python to control the system and chart data, as well as converters and drivers with serial I2C bus interface connectivity to read/write data from/to Puma560 robot. The ideas here explained could be applicable to other physical systems with similar characteristics and number of variables.

Keywords: **Control Education, Instrumentation Education, Interconnection Technologies, Remote Laboratory, Raspberry Pi, Data Acquisition.**

---

1. INTRODUCTION

Although simulations and virtual laboratories are extremely useful tools in control education, it is often difficult to provide the students with enough flexibility or accuracy to understand concepts such as the impact of control algorithms and its parameters, the effect of signal acquisition noise or the effect of a fit sampling time (Ma et al., 2006).

In the search of a greater availability of physical systems for practical educational tasks, the control education area has, for some years, made an effort in the application of new technologies. As a result, the research line of remote laboratories has received special attention for its purpose to share and improve the access to devices through the Internet in order to enable hands-on sessions on real equipment located at different locations (Aktan et al. 1996; Antsaklis et al, 1999; Ruano et al., 2016; Guzmán et al., 2010).

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, it was found that the LRA-ULE remote laboratory needs to be upgraded to achieve the following aims:

To provide a more flexible and extensible infrastructure that enables the connection of more devices and simultaneous users.

To provide an application programming interface (API) that enables the easy and rapid development of adapters for different devices and communication technologies.

To enable the usability of the web from mobile devices, which account for around the 50% of the Internet traffic nowadays.

To update the technologies used in the laboratory to the novel Internet standards. The use of appropriate technologies and frameworks enables a greater modularization and the development of user interfaces that provide the student with easier-to-use visually- and data-rich experiments.

One of the new devices that is considered to add in that process is a Programmable Universal Machine for Assembly (Puma560) robot. This paper proposes an inexpensive solution to bring to life an outdated Puma560 robot with two final goals: using the Puma560 motors in hands-on sessions (teaching instrumentation and control technologies) and enabling a future connection to the LRA-ULE remote laboratory. With regard to its educational use, the first aim is to use the Puma560 robot for the "Industrial Instrumentation" subject, so that students understand concepts as signal noise, floating voltages, sampling time,... using a flipped classroom setting. A second aim is to use only one controller to monitor and manage all the motors of the Puma560 robot in a coordinated manner.

Our assumption is that the low-cost open source hardware is much more powerful than the old Puma560 controller and can be used to provide that connectivity. Initially, an experience based on Arduino and Simulink was designed and developed for the control of the Puma560 robot. Although it was possible to properly operate with one motor, the scalability was difficult, since an Arduino microcontroller would be needed to control each motor of the 6 axes of the robot (Blanco et al. 2016). For that reason, in this paper, we present a different approach for the upgrade process of the outdated technology of the robot DC motors.

This paper is structured as follows: section 2 presents the education targets; section 3 describes the physical system (Puma560), controller (Raspberry Pi3) and data acquisition hardware that are used in this experience; section 4 explains the proposed structure for acquisition, control and monitoring; section 5 explains the practical tasks students have to complete and finally, section 6 exposes conclusions and future work.

2. EDUCATIONAL TARGETS

The suitability of DC motor units for a wide variety of control-related experiments is well-known and many educational products, such as Feedback MS-150, have been used extensively for this purpose. Indeed, it is used with our students to practice control-related topics in hands-on and remote sessions (Fuertes et al., 2013; Fuertes et al., 2012; Reguera et al., 2015).

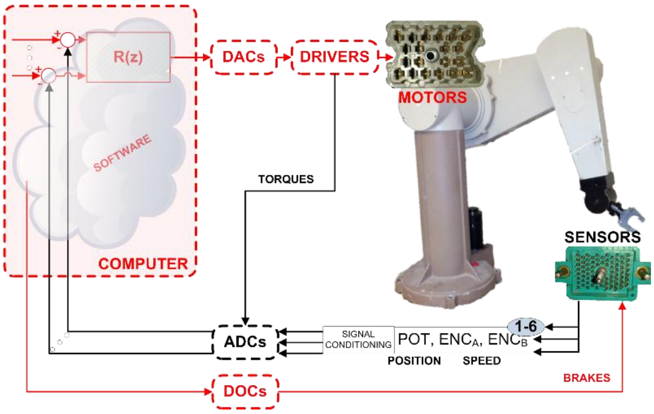

Fig. 1. Puma560 Robot control and monitoring structure.

However, the upgrade of an available outdated robot, the Puma560, built during the eighties (1982) and with an old controller but with its motors properly working, seemed a valuable opportunity to achieve a set of educational targets:

Educational tasks using real-world systems such as robotic arms give students a chance to learn how to make signal acquisition and design control systems for the real world (Dormido et al., 2008).

Courses of "Industrial Instrumentation" and "Control Engineering" are being taught at the University of Leon. Their expected learning results include: "ability to design electronic and industrial instrumentation systems", "basics of analog and digital signal processing", "knowing the components of data acquisition systems: signal conditioning, Analog-Input Card (AIC), Analog-Output Card (AOC), Digital-Input Cards (DIC) and Digital-Output Cards (DOC)" and "performing computer control of discrete-time systems by classic control and state space methods". So the use of Puma560 is very useful to understand how the appropriate acquisition and processing of signals must be done in real physical systems, as well as the control. This is a very suitable task to be done with students using the flipped classroom teaching model.

The upgrade of the Puma560 allows to expand the control practical tasks that students usually perform, e.g., enabling a wide range of control design experiences (algorithm selection, parameterization, selection of different inputs or cascade loop structures, MIMO and SISO systems are possible too) or providing do-it-yourself acquisition, control or system identification contents to the students.

The use of low-cost hardware (signal acquisition and control) elements stimulates the students to apply their skills to solve other problems with their own physical control structures, and therefore strengthens the acquisition of the core competences of their degrees.

Students must build the signal acquisition, control and monitoring structure. A faculty member provides an initial basic design of the structure (Fig. 1) and also the documentation of all necessary hardware. Students must select the signals to read, they must design and implement the signal filtering and conditioning circuits, they must select the signals to write and the control algorithms to be used. In short, it is a signal acquisition, control and monitoring structure built by students for students (supervised by a faculty member). This is a flipped+cooperative learning approach.

The development of a compact, rugged, and open structure that can be integrated, in the future, into LRA-ULE remote laboratory makes it possible to work with real data useful for further activities.

3. PHYSICAL SYSTEM AND LOW-COST HARDWARE

3.1 Physical system

The physical system is a Puma560 robot. This robot has 6 joints (3 in the body and 3 in the wrist). This robot has a brushed DC motor for each joint and its specifications are shown in Table 1.

Table 1. Motor specification (Chao et al., 2009)

Parameter	Motor 1-3 body	Motor 4-6 wrist
Current (A)	5.3	1.5
Voltage (V)	40	32
Power output (W)	150	30
Speed (RPM)	1200	2350
Torque constant (Kg*cm/A)	2.58	0.973
Voltage Constant (V/Krpm)	26.5	10
Sin/cos encoder slots	250	250
Circular potentiometer	Yes	Yes

The upgrade of the robot involves using a new power source, as well as sampling/reading all signals of the system (see Fig. 2): potentiometer, index and encoder signals (30mV sin/cos quadrature encoder signals) to measure position and velocity.

3.2 Low-cost hardware selection. Computer.

Nowadays, there are many types of low-cost open source controller boards based on AVR or ARM processors. There are also compact computers with extended I/O functionality. All these systems, with a viable cost for students, can be used for prototyping control and monitoring platforms: Arduino, Raspberry Pi, LattePanda, Intel Edison, ...

Arduino y Raspberry Pi have the lowest cost (generally less than 50€), but the Raspberry Pi is a low-cost, credit card-sized computer (not a microcontroller) used by a global community of educators. There are many free resources and projects available. Therefore, students have enough documentation available to take advantage of the functionality of this small computer.

Arduino and Raspberry were evaluated and the conclusions were: Arduino can be very useful for signal acquisition and conditioning (Arduino working as an oscilloscope) and Raspberry can be very useful because of it has a preinstalled network interface and larger computing capacity than Arduino.

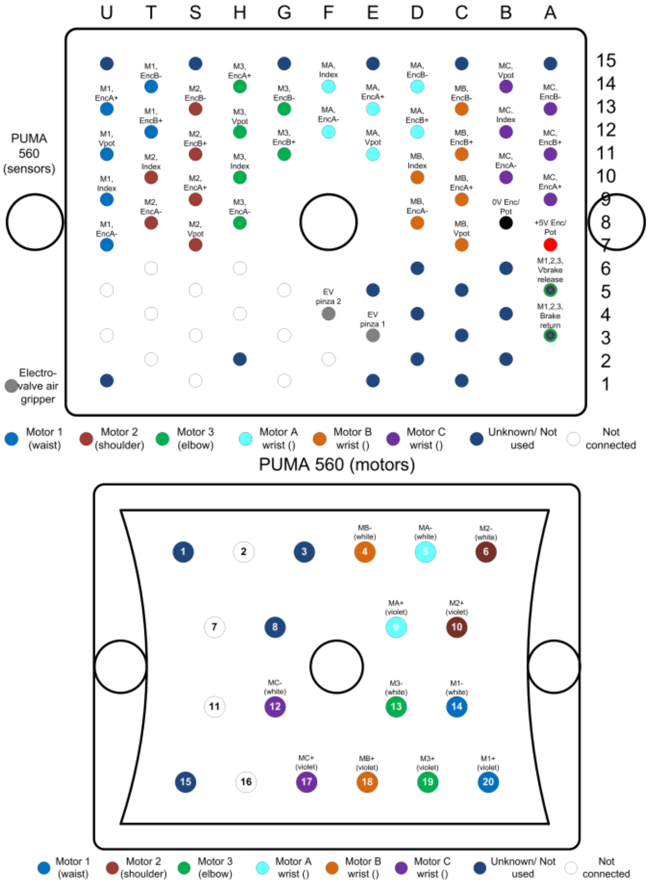

Fig. 2. Sensors and motors connector. Puma560.

The Raspberry Pi3 is the latest model and the version recommended for new projects. The greatest advantage of this device is to provide the same connectivity of a regular computer with a low price. It is used to read/write signals from/to acquisition boards and manage the control of the motors too.

This model offers support for multi-threaded operation (4 processing cores) and, due to this feature, it is 2.8 times faster in single-threaded operations than the previous model B+ and 2 times faster in multi-threaded operations than the model Pi2.

A Python package is generally used to access the Raspberry Pi GPIO (General Input Output Ports). It is important to keep in mind that accessing to GPIO is a CPU bottleneck, and its speed is relevant when used to read/write signals. The current model is 5.7 times faster than model B+ and 1.74 times faster than model Pi2 (the maximum rate achieved by a Python when toggling a pin is 344.4KHz) (Upton and Adams, 2016).

Python is an interesting option for program development for Raspberry Pi. Python syntax is very clean; it is a high level and dynamically-typed language and it is hence suitable for rapid prototyping. The main disadvantage is that Python is an interpreted language, so program execution might be significantly slower than with compiled languages such as C.

Raspbian is the Raspberry Pi Foundation’s official supported operating system. It is a Linux distribution based on Debian. It comes with Python pre-installed and students can use Python through IDLE (Python development environment) after the operating system is written on microSD card.

3.3 Hardware selection. Motor driver and power source.

A motor driver is needed to provide power to Puma560 motors by means of a specific connector (see Fig. 3). The ESCON Module 50/5 DC + ESCON Module Motherboard by Maxon Motor (one per joint) has been chosen to replace the obsolete driver. This driver was chosen due to its operating voltage and current (50V; 5A), its operating modes and its ability to measure the current of the motors (that subsequently can be used in torque control loop if required). These drivers have the higher cost and an external I2C AOC and DOC hardware is also necessary, amounting to around 250€. This driver is not the most inexpensive but it is a flexible option for the future

The Puma560 robot needs a new power source, but we want to avoid damage to the Puma560 robot´s motors caused by students, so we use PC power sources (±12 Vcc, +5 Vcc). This type of power source has a current limitation and offers a voltage about 3 times less than nominal voltage of the Puma560 robot´s motors, so these motors have a much slower dynamics, which means the current consumption and possible damages are limited. In addition to that, one more restriction is added: it is forced that ESCON Module provides a maximum speed less than 40 rpm. In this way the motors can be governed in a safely way, limiting or avoiding, any damages that students can lead with their programs.

3.4 Low-cost hardware selection. Data acquisition.

As shown in Fig. 2, it is necessary to read 3 analog signals from each motor: potentiometer signal, encoder signal A (differential mode), and encoder signal B (differential mode). As a result, the total number of input analog signals is 6x3=18. Considering that number, the appropriate choice for data acquisition would be a Raspberry Pi3 + external acquisition hardware.

The index signal could be the 4th analog signal to be read from each motor due to it improves the potentiometer precision. Although, this signal can be read as digital to make the control of each motor, however, from a learning point of view, it is interesting that signal can be read as analog by the students of "Industrial Instrumentation" because of the difficulty of its acquisition.

The proposed acquisition boards are connected to Raspberry Pi using the I2C Bus. I2C is a multi-device serial bus and its main advantage is more than one device can be connected without using up additional pins on the header.

AICs have I2C communication and 5 channels with a resolution of 10 bit (0-10V signals can be read). This card has a microcontroller (PIC 18F13K22) that establishes the connection between the I2C bus and Analog-to-Digital Converter (ADC). It ensures that the data is updated in the registers with a frequency of 80 Hz per. To read all 18 signals (these AICs are not able to make the acquisition of signal index as analog), 4 analog input cards are needed.

In this specific case, the main advantage of the used acquisition boards is the scalability provided to the system, i.e., additional signals can be read/written (such as a new motor) simply by adding a new acquisition board and modifying the Python source code to include the new address of the connected elements.

The clock frequency of Raspberry Pi3 is 1.2GHz, the I2C bus default speed is 100KHz and the ADC ensures that the data is updated in the registers with a frequency of 80Hz per channel, so 12.5ms is the minimum sampling time that can be used (ADC is a bottleneck since is shared by all channels). The sampling time mainly depends on the cycle time of the program running in the Raspberry Pi and the number of signals read and written. Other available acquisition boards provide lower sampling times, but scalability has been preferred over performance.

An important issue is that the maximum speed of the motors is achieved for voltages in the range of 30-40V. If that voltage is kept at lower values, sampling time could be longer (this is the case we have when switched PC power sources of ±12 Vcc are used). However, it must be emphasized that the achievement of fast sampling rates in an industrial environment does not have as much educational value as the proper understanding of the whole process of signal acquisition and conditioning with limited resources.

Motors 1-3 (body joints) incorporate a 24V electromagnetic brakes which must be released before operating these motors, so 3 additional digital output signals must be considered and external I2C DOC hardware is necessary. In addition to the previous hardware, an I2C repeater for Raspberry Pi is necessary. This repeater translates from 3.3V to 5V for GPIO ports SDA and SCL.

Altogether, considering previous computations, 24 analog input signals are read, and 6 analog and 3 digital output signals are written. These numbers mean that 5 analog input cards and 2 analog output cards are needed to read/write all analog signals along with 1 digital output card to write all digital signals. It would have the necessary hardware to read the index signal if more precision is needed.

The total cost for 6 motors (6 drivers + 5 AICs + 2 AOCs + 1 DOC + 1 repeater) amounts to 1716€ (1320€ for the motor drivers + 396€ for the rest of the hardware). Due to the limited budget, initially, the necessary hardware to 2 motors is acquired. Note that hardware elements used are I2C, so the solution is easily scalable: so more hardware elements can be connected as soon as the budget is available.

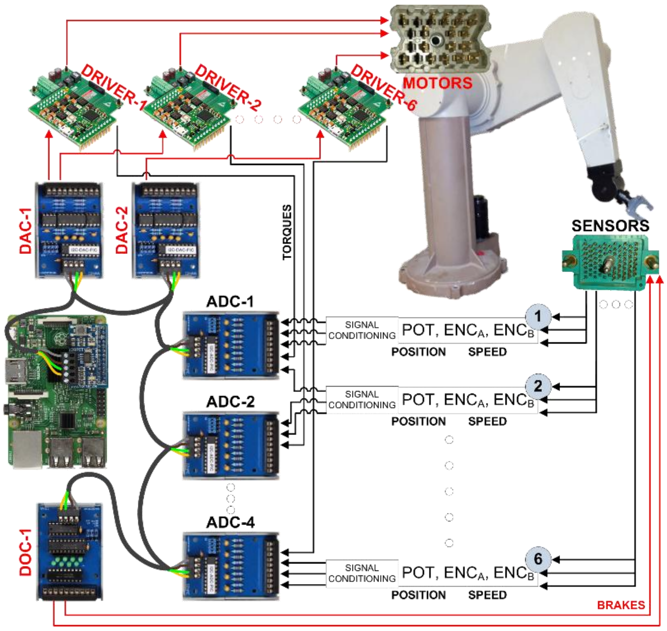

Fig. 4. Connections to the control structure.

4. CONTROL AND MONITORING STRUCTURE

The initial control structure (Fig. 4) is focused on the mentioned educational targets: upgrading the system, enabling configurable practices and providing a low-cost solution with easy signal acquisition.

The advantages of this structure and the Puma560 are:

There are 6 motors and their sensors, so we have 6 SISO different systems and 57 MIMO different systems (number of possible combinations by taking 2 motors from 6 motors + combinations by taking 3 motors from 6 motors + ...) to implement closed-loop control experiments.

I2C serial bus is used for attaching the acquisition cards (lower-speed cards) to the main controller. This bus allows the structure to be scalable.

The motors can be powered using a square signal (motors turn right and left alternatively) so that students can read feedback information from encoder and potentiometer and index signals.

The students can design and implement the signal conditioning circuit (encoder signals need signal filtering, amplification and offsetting).

The students can also test a closed-loop position (or speed or torque) control of one motor using a discrete control algorithm, make PID tuning experiences or perform system identification experiments (e.g., Strejc and Kupfmuller methods, ARX, ARMAX or OE models).

Students of the courses "Industrial Instrumentation" and "Control Engineering" at the University of Leon have to understand how appropriate acquisition and processing of signals is done in real physical systems, i.e., all the stages needed for proper conditioning of the robot sensor signals. So it is necessary to measure noisy low-level signals from the robot encoders to be used in a digital system (Raspberry Pi3). The signal acquisition structure needs to be easy and clear, so that students understand concepts as sampling, jitter, signal filtering, aliasing and other similar concepts.

5. STUDENT TASKS

Students build the control and monitoring structure: they must design the signal filtering and conditioning circuits, configure acquisition boards and code a Python program for signal acquisition, control and monitoring of the Puma560 motors.

5.1 Signal filtering and conditioning.

Students of the subject "Industrial Instrumentation" will benefit greatly from this task. They must design a circuit to read and adapt signals from one motor (encoder A-B and potentiometer signals), considering signal noise, offset, etc. The first stage of the task is focused on the simulation of the circuit in PSpice Schematics. This circuit must be positively evaluated by a faculty member.

Then, students implement the designed circuit in a breadboard so that it can be connected to the robot. To check its correct operation, one of the motors is powered by ±12V square signal and the signal of the encoder and index are visualized in the oscilloscope and acquired by student´s circuit, with results similar to shown in Fig. 5. An Arduino Mega and C# is used to acquire the index as analog signal (only for viewing, not for use in control loop).

An RC passive low-pass filter with a cutoff frequency of 1 Hz (R=4.7 MΩ; C=33nF) can be used with the potentiometer signals and a high-pass filter with a cutoff frecuency of 3200 Hz (R=2200Ω; C=22µF) is used with the index signals. With regard to the encoders, initial filtering can be performed with an RC passive low-pass filter with a cutoff frequency of 60 Hz (R=68 KΩ; C=39nF) to suppress noise. This way, the signal can either be properly represented graphically (Fig. 5).

Students, in the case of encoder A-B signals, may choose to represent the signal waveform or just keep its phase and frequency (when signals are transformed to a 0-10V square signals), that is what conveys the direction of rotation and the velocity of the motor.

From a computational perspective, if the aim is to control all the motors simultaneously, the square wave signals can be sampled using a longer sampling period (needed if simultaneous sampling with a relatively constant sampling time is required). Nevertheless, from an educational perspective, it is quite useful that students perform pre-filtering of the signal of at least one of the motor encoders in order to observe the effect of the correct selection of sampling period in the representation of the frequency and shape of the sampled signal.

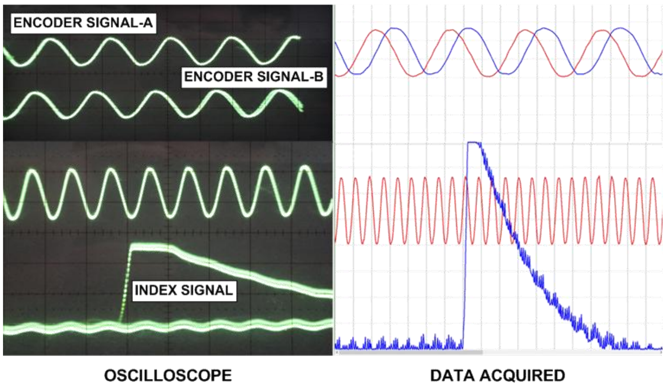

Fig. 5. Encoder A-B and index signals in oscilloscope and data acquired by student´s circuit (1 ms/div).

Students should also be able to read the conditioned signals in a 0-10V range in a program developed in Python that is run in a Raspberry Pi3. Potentiometer and encoder signals include noise so they must be filtered. Encoder signals have a ±30mV range and the potentiometer signals, 0-5V, i.e., both must be converted into 0-10V analog signals (AICs).

5.2 Configuration acquisition boards and motor drivers.

Students must configure the corresponding acquisition board channels. They must design the address mapping, define the range and set the I2C bus address of each one so they can be written and read from Python language. The motor drivers need to be configured using the manufacturer´s software with the appropriate voltage and current values. Students need to acquire have in-depth knowledge of that software.

5.3 Control software and data charting (Python code).

There is extensive documentation about signal communication with I2C using Python, students can take advantage of this previous information to code the necessary Python program (running on Raspberry) to read data from all Input cards (AIC, DIC) and to write data on Output cards (AOC, DOC). That program should include a Graphical User Interface (GUI) to use it: buttons, textboxes, checkboxes and real-time data charts. From an educational point of view, charting signals from the Puma560 robot is very important to verify correct signal acquisition and to see the different control signals. This program must also provide a button to start/stop logging data to a file, which can be very useful to analyze the data using mathematical software. This way, students can easily system identification experiments (Strejc and Kupfmuller methods, ARX, ARMAX or OE models).

Standard Python packages such as TkInter allow a student with no previous knowledge of the programming language to quick develop a GUI that shows graphically the evolution of the acquired signals. The collected data is plotted with

Matplotlib (Python 2D library), which has been selected because our students are used to work with Matlab and this Python library has a similar syntax.

The students of "Control Techniques" implement 3 control algorithms in Python: on-off, PID and dead-beat using a SISO system. The response obtained using a PI control for a motor is showed in Fig. 6. On the other hand, students of the "Industrial Instrumentation" are focused on the previous task (signal filtering and data acquisition).

In Fig. 6, only the potentiometer signal was used to perform the control, and even using that signal, students could observe the effect of a noisy signal (potentiometer) in the control-loop, also the effect of having a greater or lesser sample period (oscillations in the final value) and the dynamics of these types of motors

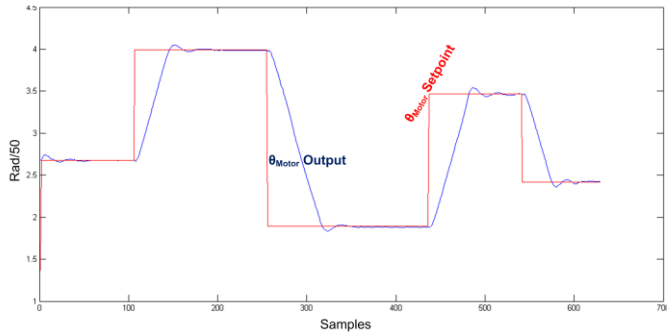

Fig. 6. PI control on a Puma560 motor (20 ms/Sample).

6. CONCLUSIONS AND FUTURE WORK

An outdated and non-operational Puma560 robot system has been brought to life and upgraded using low-cost but scalable technologies with a project-based learning purpose. The concepts applicable to the structure and Raspberry Pi programming to monitor and control a robot are an attractive and intuitive way to learn for students.

The structure and hardware have been assessed with students of the “Industrial Instrumentation” and “Control Techniques” courses, both taught in the fourth year of the Engineering Degree in Industrial Electronics and Automation. Students developed the signal filtering and conditioning and configure acquisition boards on their own, taking advantage of the provided information. They installed the Raspbian operating system on their own devices, configured the network, coded a program to read, write and chart signals of the Puma560 robot and finally developed the control algorithms.

As a result, a compact, rugged, and open structure that can be integrated into LRA-ULE remote laboratory has been developed. For that, students are installing Flask on the computer to develop a web service to read values from Puma560 sensors and to write values to control one motor. Flask is a Python package, chosen because of its simplicity, that can be used to develop of a web server in the Raspberry Pi that provides access to the input/output variables using a REST (Representational State Transfer) service.

Future work will be focused on the integration of the system in the LRA-ULE remote laboratory. For that purpose, the use of lightweight communication between the system and the laboratory is needed.

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*, 19.

Blanco, D., Alonso, S. and Domínguez, M., Cascade control of the PUMA560 motor using Simulink and Arduino, *Open Conference on Future Trends in Robotics. ROBOCITY16*, Madrid. Spain. 2016

Chao, J., Fernandes, F., Lie, D. and Tanis, J., Interfacing Motors of PUMA560 Robot with a PC-based Controller. *ECE4007 Senior Design Project. Section L01, Yellow PUMA Team*. 2009

Dormido, R., Vargas, H., Duro, N., Sanchez, J., Dormido-Canto, S., Farias, G., Esquembre, F. and Dormido, S., Development of a web-based control laboratory for automation technicians: The three-tank system. *IEEE Transactions on Education*, 51(1), 35-44. 2008

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.

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

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., García, D., Domínguez, M., Prada, M.A., Alonso, S. A Low-Cost Open Source Hardware in Control Education. Case Study: Arduino-Feedback MS-150. *3rd IFAC Workshop on Internet Based Control Education IBCE15*, Brescia, Italy 2015.

Upton, E. and Adams, J. Raspberry Pi3 is out now!, specs, benchmarks & more. *The MagPi. The official Raspberry Pi Magazin*e, Issue 43 10-13. March 2016.

Ruano, I., Gámez, J., Gómez, J. Laboratorio Web SCORM de Control PID con Integración Avanzada. *RIAI, Vol. 13, Issue 4, 472–483*. December 2016.

Guzmán, J.L., Domínguez, M., Berenguel, M., Fuertes, J.J., Rodríguez, F., Reguera, P. Entornos de experimentación para la Enseñanza de Conceptos Básicos de Modelado y Control. *RIAI*, Vol. 7, Issue 1, 10-22. January 2010.