A Low-Cost Matlab-Based Educational Platform for Teaching Robotics

J.J. Castañeda, A.F. Ruiz-Olaya, W. Acuña, A. Molano

Abstract— **Robotics courses play a central role in the electronic engineering curriculum. Those courses provide students with knowledge and skill in multiple aspects of the design, simulation, implementation and operation of systems using robotics technologies for applications in areas such as the industrial, medical, services, among others. This work describes the implementation and application of a Matlab-based platform using low-cost technologies as an educational tool to be included in robotics courses. The platform comprises an Arduino as a target for hardware in the loop (HIL) simulation using Matlab and a 6 Degree of Freedom (DoF) articulated robotic arm. The HIL simulation system differs from computer simulation in such a way that it involves actual hardware and permits controlling the real world actuators and sensors. It was carried out set of experiments aimed to evaluate the platform in the field of human-robot interaction; specifically, through an application to control the robotic arm to follow movements of a human upper limb. Root Mean Square Error (RMSE) was used to correlate movement provided by user and the actual movement executed by the robotic arm. Obtained mean measurement errors were RMSE of 4.7 ° for static validation and 5.3 ° for dynamic tests.**

Keywords—Educational Robotics; Hardware in the loop simulation; Arduino; Inertial sensors; Joint angle amplitude

I. INTRODUCTION

In many cases, the most effective way to develop a control, mechatronic or robotic system is to connect it to the real hardware or plant, which could be implemented using Hardware-in-the-loop (HIL) simulation. HIL can be viewed as a synergistic combination of physical and virtual prototyping ("modelling and simulation"), [1]. In hardware-in-the-loop simulation setup of a system, part of the simulation loop includes computer software, and other part includes the actual physical hardware systems. The main advantage of HIL simulation refers to the "actual" validation of the system.

HIL simulation has been widely used in the design of systems for several engineering areas that include control [2], automobile [1], robotics [3], among others. For electronic engineering teaching, HIL is a powerful tool and have been presented many papers that use it [4-6]. HIL simulations support the robotics teaching process by enabling the professor to illustrate concepts about sensors data acquisition, control of actuators, modelling of dynamic systems and development of control algorithms under different operating conditions. However, traditionally HIL simulations require high cost hardware and software, and some of more known providers are National Instruments [7], DSpace [8], Quanser [9] and Mathworks [10]. For instance, Simulink Real-Time is the HIL and prototype design tool provided by Mathworks that could be used in real-time testing [11]. It is required C compilers such as Microsoft Visual C++ or WATCOM C/C++. Furthermore, it is require specialized data acquisition cards or compatible hardware, both very expensive.

In literature, few works have been presented that implement HIL using inexpensive hardware [12]. With the advent of low-cost robotic technologies (i.e. sensors, actuators, controllers, etc.) and hardware support of simulation software packages as Matlab, an affordable computer-aided teaching tool could be extended in laboratory practices for robotics education. MATLAB and Simulink support a set of hardware from well-known manufacturers that includes National Instruments, Texas Instruments, Diamond Systems, among others, and can generate executable software for them directly from Simulink models. Recently, MATLAB and Simulink support low-cost hardware such as Arduino, Raspberry, LEGO, among others that can also be used to implement HIL simulation and practice experiments in robotics courses. In this case, Matlab/Simulink directly generates executable code for them and permits controlling the real world actuators and sensors by a hardware-in-the-simulation loop. The major characteristic is that the real-time executable code can be generated by a friendly graphical tool such as Simulink to run on the target controller.

This paper focuses on implementation and application of a Matlab-based platform using low-cost technologies, that includes a HIL system and a robotic arm, as an educational tool to be included in robotics courses. The platform comprises an Arduino as a target for HIL simulation to control a 6 DoF articulated robotic arm. The field of human-robot interaction (HRI) is a significant area of robotics to create interfaces that enable natural and effective modes of interaction with robotic technologies. It was carried out a validation process of the developed platform as a practical demonstration for teaching HRI.

The remainder of the paper is organized as follows. Section II describes the educational platform. Section III explains the experimental methods of validation, and compares the data obtained from experiments with a well-known system. Finally, Section IV presents conclusions and explains how to integrate this educational platform into undergraduate robotic courses.

J.J. Castañeda, Bioengineering research group, Universidad Antonio Nariño, Bogotá, Colombia (E-mail: jhocastaneda@uan.edu.co)

A.F. Ruiz-Olaya, Bioengineering research group, Universidad Antonio Nariño, Bogotá, Colombia (E-mail: andresru@uan.edu.co)

W. Acuña, Bioinstrumentation and control research group, Universidad Antonio Nariño, Bogotá, Colombia (E-mail: wilber.a.b@uan.edu.co)

A. Molano, Bioinstrumentation and control research group, Universidad Antonio Nariño, Bogotá, Colombia (E-mail: andres.molano@uan.edu.co)

978-1-5090-3787-2/16/$31.00 ©2016 IEEE

II. MATLAB-BASED EDUCATIONAL PLATFORM

A. Block Diagram of the System

Fig. 1 presents a block diagram of the system. As a target, it was selected the open-source hardware Arduino, model MEGA2560. Furthermore, it was designed and built a 6 DoF articulated robotic arm powered by servomotors.

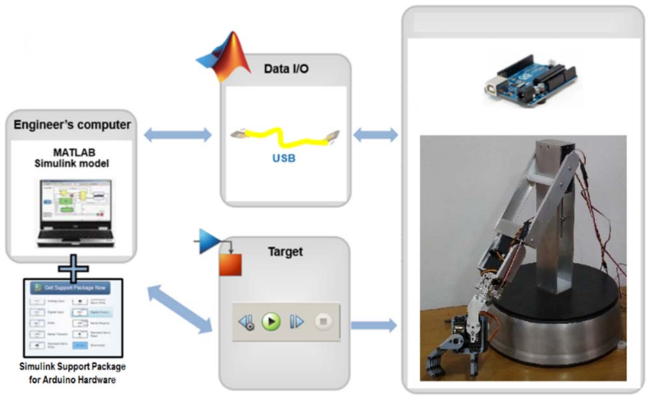

Figure 1. Block diagram of the Matlab-based educational platform

B. Hardware-in-the-loop Simulation with Matlab/Simulink and Arduino

Since version R2012b of MATLAB, Simulink provides hardware support packages for third-party products such as Arduino, Raspberry, and LEGO Mindstorms. The platform for hardware-in-the-loop simulation using Arduino is a low-cost solution to experimental education of robotics. It is required to download drivers from Internet, selecting the appropriate product (Arduino Mega was used with this work).

Once completed the installation procedure, the "Simulink Arduinolib" is automatically embedded into the Simulink Library Browser. This graphical library provides functionality for acquiring digital and analog inputs, controlling digital and PWM outputs, serial communication capabilities, among others (see Fig. 2).

The Simulink model may include all libraries of Simulink and computational options to create complex models. Finally, it is required to set up model to run on the target (Arduino MEGA2560 in this case). A powerful feature provided by Simulink to support Arduino hardware is the real-time connection for tuning parameters of the model.

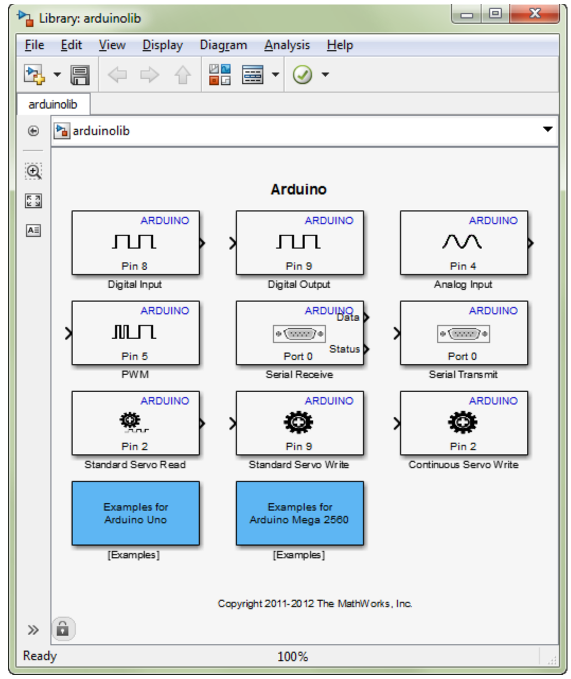

Figure 2. Simulink Arduinolib.

C. Six DoF Robotic Arm

A robotic arm is a programmable mechatronic structure, similar to the human arm (anthropomorphic). It was designed and built a low-cost robotic arm that includes 6 DoF. The mechanical structure was built with aluminum and the segments were interconnected through articulated joints that allow rotational movements. The articulated robot with 6 DoF allow complex movements similar to these of human arm.

The robotic arm was implemented using 6 actuators (servo motors) describe in Table I.

TABLE I. ACTUATORS FOR A 6 DOF ROBOTIC ARM

Degree of Freedom	Servomotor	Torque (kg-cm)	Robotic Arm Movement
1	HITEC HS-805BB	24.7	Shoulder flexion-extension
2	HITEC HS-645MG	9.6	Elbow flexion-extension
3	HITEC HS-485HB	6.0	Forearm pronation-supination
4	FUTABA S3003	4.1	Hand open-close
5	POWER HD 1501MG	7.0	Wrist flexion-extension
6	HITEC HS-755HB	13.2	Base rotation

Fig. 3 shows the articulated robotic arm, and indicates actuators of each degree of freedom (DoF).

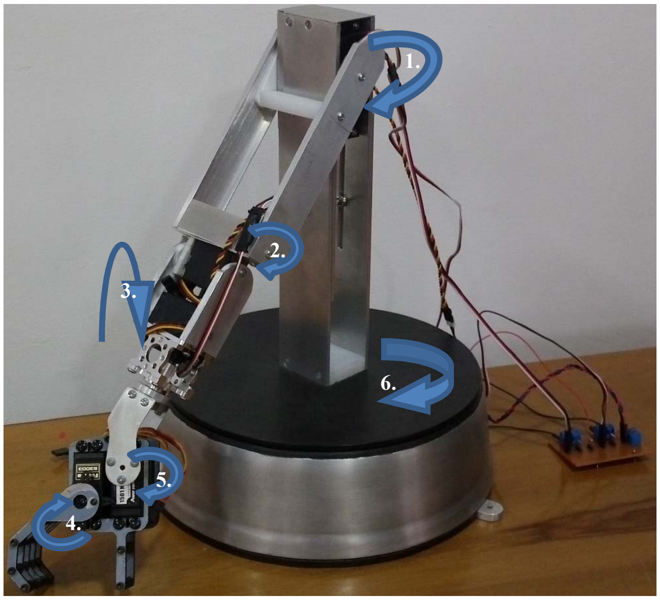

Figure 3. Low-cost 6 DoF articulated robotic arm powered by servomotors.

III. EXPERIMENTAL METHODS

As a case study of validation of the educational platform, it was carried out a set of experiments as a practical demonstration for teaching HRI, which is a significant area of robotics courses. Specifically, it was used HIL simulation to control angular position of two degrees of freedom of the robotic arm, specifically shoulder flexion-extension and elbow flexion-extension. The system was performed on the HIL setup. The reference angle or setpoint come from joint angle of human upper limb (shoulder and elbow) calculated using inertial sensors (accelerometers). First at all, next it is presented experimental methods to obtain the human joint angle. Later, it is describe the Simulink model. Finally, the experimental protocol and results are presented.

A. Estimation of Human Joint Angle

Advances in MEMS (Microelectromechanical Systems) enable systems based on inertial sensors (accelerometers, gyroscopes) for motion capture [13]. An accelerometer measures the acceleration that the sensor is subject to. An accelerometer is a device that measures its proper acceleration that is the physical acceleration experienced by it relative to a free-fall, or inertial, observer who is momentarily at rest relative to the object being measured.

It was used the accelerometer MMA6361L from Freescale Semiconductor which is a tri-axial device that could measure ±1.5g. In real applications obtaining the angular position is not simple due to the noise in signals.

Accelerometer data is a vector, having 3 axes (x, y, and z). This vector permits to measure gravity acceleration and any other acceleration the device is subject to. For instance, Fig. 4 shows an accelerometer that have axis x and z used to calculate a tilt. Fig. 4 (middle) shows the element with a tilt (θ) and Fig. 4 (right) shows components provided by accelerometer, according to gravity vector.

The acceleration in the x-axis and z-axis is calculated by the following equations:

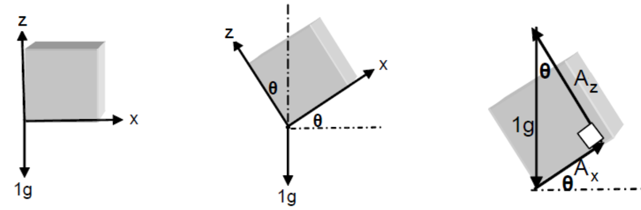

Figure 4. Accelerometer used to calculate a tilt (angle θ).

$$A_x = \sin\theta \tag{1}$$

$$A_z = \cos\theta \tag{2}$$

Thus, the angle θ could be calculated using following equations:

$$\tan\theta = \frac{A_x}{A_z} \tag{3}$$

$$\theta = tan^{-1}\left(\frac{A_x}{A_z}\right) \times \frac{180}{\pi} \tag{4}$$

where the angle θ is in degrees.

However, the angle resolved is correct when the accelerometer measure the gravity only, taking into account that the resolution is biased when the accelerometer accelerate.

In order to calculate the angles of the accelerometer in three dimensions it could be defined the pitch (ρ) as the angle of the x-axis relative to ground and roll (ϕ) as the angle of the y-axis relative to the ground (see Fig. 5).

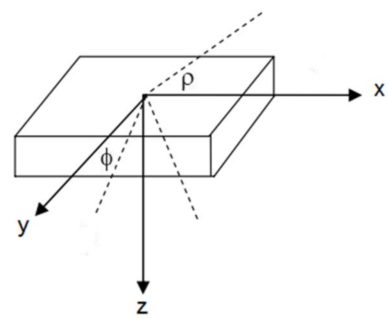

Figure 5. Accelerometer used to calculate pitch and roll angles.

The pitch and roll angles could be calculated using following equations:

$$\rho = tan^{-1}\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \times \frac{180}{\pi} \tag{5}$$

$$\phi = tan^{-1}\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right) \times \frac{180}{\pi} \tag{6}$$

B. Simulink model

The Simulink model includes blocks that provide capabilities for:

- Acquiring analog information of sensors and digitalizing it. Specifically, it was recorded four analog signals: two from each accelerometer (axis-x and axis-z).
- Filtering and processing information of sensors and to generating a command signal to control actuators.
- Controlling angular position of servomotors, through PWM. It was used 2 actuators.

Fig.6 shows the Simulink model. Later it was run model on target (Arduino Mega).

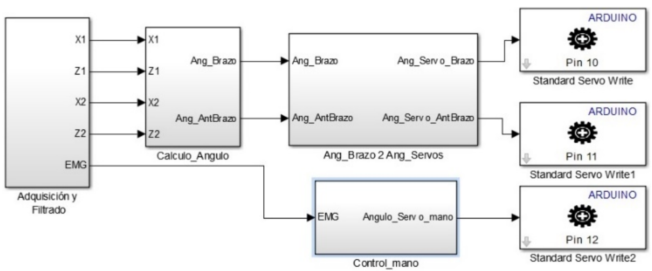

Figure 6. Simulink model of an application to control a robotic arm to follow movements of a human upper limb.

The Simulink model consists of four main blocks: Acquisition and filtering, calculation angle, conversion of estimated angle to the angle of the servo motors and hand control.

The first block "Acquisition and Filtering" contain required filters for conditioning analog input signals. It was used digital low-pass Bessel second order filters, whose cutoff frequency is 20Hz.

Fig. 7 shows Simulink blocks for calculation of joint angle from accelerometer signals, according to Equation 4.

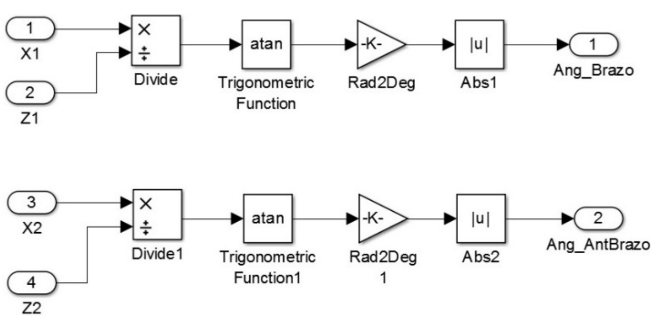

Figure 7. Calculation of joint angle from accelerometer signals.

The block "Conversion" in the Simulink model permits to calculate control commands for servomotors.

C. Protocol

a) Instrumentation:

One healthy subject participated in the experimental protocol. Two accelerometers were used to monitor the flexion-extension angle of elbow and shoulder joints. Accelerometer 1 was attached to arm and accelerometer 2 was attached to forearm, such a way that axis-x of accelerometers coincide with sagittal plane of flexion-extension movements. Estimation of joint angle of elbow and shoulder follow procedure described below.

For validation, the joint angles estimated from accelerometers was compared to joint angles of the robotic arm (elbow and shoulder), obtained using a vision-based motion capture system. These systems are widely used to obtain joint kinematic data. The vision-based motion capture system used for validation consists of high speed cameras, and software to capture and analyze the collected information. The vision-based motion capture system for acquiring joint kinematics data combines 3 cameras Basler SCA640-70 GC GIGE, passive reflective markers and commercially-available motion analysis software. The software MaxTraq 3D from Innovision Systems Inc. was used to capture and digitize the data. Four reflective markers were attached on specific segments of the robotic arm (see Fig. 8). These markers defined a system with 2 rigid segments, arm and forearm.

It was aligned the axis from the vison-based motion capture reference system to axes to be measured from accelerometers.

Figure 8. Movement measurement of human upper limb and robotic arm.

Both systems were synchronously recorded during flexion-extension elbow and shoulder movements, with a sampling frequency of 30 Hz.

b) Quasi-Static Flexion-Extension Test:

First of all, starting from maximum elbow extension, the forearm was bent to maximum flexion through 4 steps. In each step, the posture was held to rest for about 10 seconds and the average value of the recorded data was computed within the last 5 seconds.

Later, procedure was executed for shoulder extension-flexion. Fig. 9 shows obtained data.

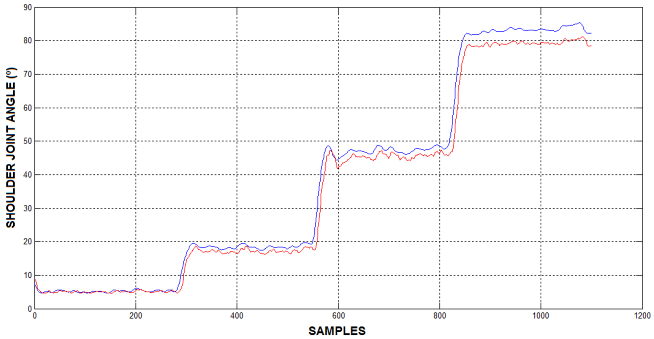

Figure 9. Joint angle estimated from accelerometers (red line) and provided by the vision-based motion capture system (blue line) during quasi-static shoulder flexion-extension.

c) Dynamic Test:

First of all, subject executed elbow slow-speed flexion-extension cyclical movements for 30 seconds. Later, subject executed shoulder slow-speed flexion-extension cyclical movements for 30 seconds.

Fig. 10 presents joint angle data of one section for a subject executing a slow-speed cyclical flexion-extension shoulder movement.

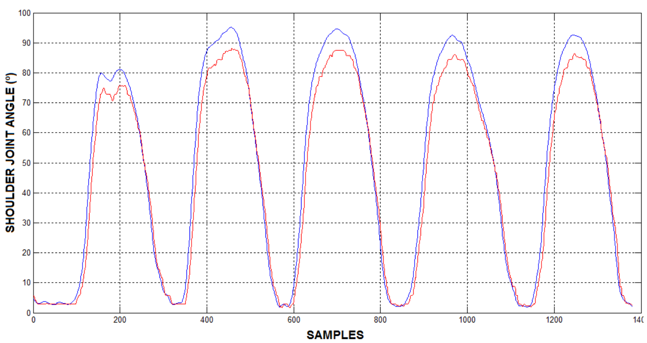

Figure 10. Joint angle estimated from accelerometers (red line) and provided by the vision-based motion capture system (blue line) during cyclical flexion-extension shoulder movements.

D. Results

Data analysis was carried out by relating the angle estimated from accelerometers with angles provided by the vision-based motion capture system (reference). The root mean square error (RMSE) was used to quantify the difference from the value of the reference and the value estimated, as can be seen on equations (7) and (8).

$$e(n) = angle_{reference} - angle_{estimated} \quad (7)$$

$$\text{RMSE} = \sqrt{\frac{1}{k}\sum_{n=1}^{k} e^2(n)} \quad (8)$$

Overall results are reported in the Table II. Joint angle estimated from accelerometer and executed by the robotic arm consistent with data recorded using vision-based motion capture system.

TABLE II. CALCULATED RMSE.

Test	Obtained Results	
	Joint Movements	***RMSE (°)***
Quasi static	Elbow	5.3
	Shoulder	4.1
Dynamic	Elbow	7.1
	Shoulder	4.5

IV. DISCUSSION AND CONCLUSIONS

It was presented an innovative educational tool to be included in engineering courses, for teaching robotics. It consists of an experimental MATLAB/Simulink based hardware-in the- loop simulation platform and a 6 DoF robotic arm. This system was built using affordable low-cost technologies.

The platform provides interaction with the real world (sensors, actuators, plants). This is of special importance in robotics, because several times accurate mathematical models cannot be obtained, so exact Simulink models cannot be constructed and simulated. For instance, using the developed platform, students can apply, design, simulate and implement control systems that are needed in operating of robotics platforms, doing use from Simulink blocks and Matlab codes.

Furthermore, the platform can be used to enhance the knowledge about kinematics and kinetics of robotics arms, signal processing from the sensors, human-robot interaction, among other. Particularly, HRI is an important area of robotics courses focused on developing interfaces that enable effective modes of interaction with robotic technologies. Thus, a successful integration of Simulink models to run on target (Arduino) was described in an application to control a robotic arm in order to follow joint angle of human elbow and shoulder joints, as a practical demonstration of HRI. A main advantage of the system is scalability and flexibility.

ACKNOWLEDGMENT

Authors thank Universidad Antonio Nariño for support of this work.

REFERENCES

[1] H.K. Fathy, Z.S. Filipi, J. Hagena, and J.L. Stein, "Review of Hardware-in-the-Loop Simulation and Its Prospects in the Automotive Area", Modeling and Simulation for Military Applications, edited by Kevin Schum, Alex F. Sisti Proc. of SPIE Vol. 6228, 62280E, 2006.

[2] N.R. Gans, W.E. Dixon, R. Lind, and A. Kurdila, "A hardware in the loop simulation platform for vision-based control of unmanned air vehicles", Mechatronics, vol. 19, pp. 1043–1056, 2009.

[3] L. Zouari, M. Ben Ayed, and M. Abid, "A Hardware In the Loop Simulation for Electrically Driven Robot Manipulator", Proceeding of the IEEE 28th Canadian Conference on Electrical and Computer Engineering Halifax, Canada, May 3-6, pp. 1495-1501, 2015.

[4] A. Sala and J. Bondia, "Teaching Experience with Hardware-in-the-Loop Simulation", 7th IFAC Symposium on Advances in Control Education, vol. 39, Issue 6, pp. 123-128, 2006.

[5] W. Grega, "Hardware-in-the-loop simulation and its application in control education", 29th ASEE/IEEE Frontiers in Education Conference, 1999.

[6] W. Martono and R. Muhida, "Implementation of Hardware-in-the-loop Simulation (HILS) Method for Control Engineering Education", Proceedings of the 2005 Regional Conference on Engineering Education, pp. 233-238, 2005.

[7] National Instruments. http://www.hil-ni.com/en/

[8] dSPACE. https://www.dspace.com/en/inc/home.cfm

[9] Quanser. http://www.quanser.com/

[10] Mathworks. http://www.mathworks.com/

[11] Simulink Real-Time. http://www.mathworks.com/products/simulink-real-time/

[12] W.J.E. Salami and A. Albaguld, "Development of a Microcontroller-Based Control System with a Hardware-in-the- Loop (HIL) Method for Control Education using MATLAB/Simulink/xPC Target", Int. J. Engng Ed. Vol. 21, No. 5, pp. 846-854, 2005.

[13] H. Luinge, Inertial sensing of human movement, Ph.D. dissertation, University of Twente, 2002.