

## Development of Microcontroller-Based Ball and Beam Trainer Kit

Gunawan Dewantoro<sup>\*1</sup>, Deddy Susilo<sup>2</sup>, Ditya Clarisa Amanda<sup>3</sup>

Department of Electronics and Computer Engineering, Satya Wacana Christian University,  
52-60 Diponegoro Street, Salatiga, Indonesia 50711

e-mail: gunawan.dewantoro@staff.uksw.edu<sup>1</sup>, deddy.susilo@ymail.com<sup>2</sup>, ditya.clarisa@gmail.com<sup>3</sup>

### Abstract

A ball and beam trainer kit based on microcontroller was developed for teaching control system course for the sophomore students. This specially-purposed kit consists of a ball located on a beam with a fixed axle at one of its end. At the other end, a servomotor was employed to control the position of the ball by adjusting the rotation angle of the servomotor. Seven predetermined positions were set to 10, 20, 30, 40, 50, 60, and 70 cm relative to the fixed axle of the beam. The Proportional-Integral-Derivative (PID) scheme was then used to compensate the error. This kit is equipped with a user interface to configure controller coefficients, select the set points, plot the actual ball position, and display parameter values. The user interface program runs on PC or notebook connected to microcontroller via serial communications. A questionnaire-based assessment about the use of this kit was conducted by 17 students taking the course, giving a rating value of 94.12%.

**Keywords:** ball and beam, control system, PID, user interface

### 1. Introduction

Traditionally, control systems courses are offered by Electrical and Mechanical Engineering Departments all over the world. As in many other university-level technical courses, interactive tools have been very helpful for the students in order to better comprehend the substances along with formal lectures and exercises. Mostly, an interactive tool in control education provides graphical windows which employ active, dynamic, and clickable components. An interactive tool using Easy Java Simulations was implemented in the field of Control Engineering education [1]. A simulator was developed using user friendly virtual interface software to control the speed of a small size DC motor. The user is able to select and adjust the parameters of any desired controller that is defined and represented virtually [2]. Educative tools, Linear System Analysis and Design (LSAD) [3] and Interactive Learning Modules for Proportional-Integral-Derivative (ILM-PID) [4], were built to make students more active and involved in their own control engineering learning process. Excel spreadsheets for basic control education was developed to help students understand how feedback works [5]. The Ch-Control System Toolkit, a software package for the design and analysis of control system, was utilized to develop Web-based Control System Design and Analysis System (WSCDAS) and Web-based Controller/Compensator Design Module (WCCDM). These software tools were used for teaching automatic control of linear time-invariant system [6]. A control simulation for nonlinear system was applied using JAVA-enabled open source software, therefore, students have all necessary elements to practice using nonlinear system which in this case an inverted pendulum [7].

However, all of the above mentioned works were merely implemented by the use computer-based virtual tools. An actual system needs to be constructed in companion with such virtual counterpart. Microcontroller is a compact microcomputer containing a processor core, memory, input/output peripheral, timer/counter, which is able to govern the operation of any plants. Microcontrollers have also been used in learning control system with laboratory scale practical models. Low-cost digital temperature control kit was designed to implement various control strategies using microcontroller [8]. A microcontroller-based control application was designed to distribute supervisory and control tasks within a plant in industry [9]. Low-cost educational microcontroller-based tool for fuzzy logic-controlled line-following mobile robot was put into practice for teaching second-year students [10].

Ball and Beam is a widespread example of application of feedback control system and has been theoretically taught in most universities. In this paper, an ATmega 8535 microcontroller-based ball and beam simulator was built to encompass both virtual and actual tools for learning control system course. This kit consists of a long beam which can be tilted to control the displacement of a ball on it using Proportional-Integral-Derivative (PID) controller. Hardware details and the user interface implementing the controller are given in the paper.

## 2. System Modeling

A set of mathematical equations was addressed to represent the ball and beam employing appropriate physical laws. All variables contained in the ball and beam system are shown in Figure 1.

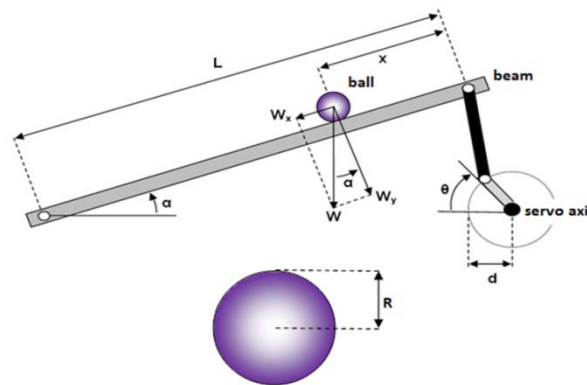


Figure 1. Ball and Beam physical model

Translational acceleration of the ball is given by:

$$\frac{d^2 x}{dt^2} = \ddot{x} \quad (1)$$

The exerted force due to translational motion,  $F_{tx}$ , is:

$$F_{tx} = m\ddot{x} \quad (2)$$

While the torque of the ball due to rotational motion,  $T_r$ , is:

$$T_r = F_{rx} R = J \frac{d\omega_b}{dt} = J \frac{d\left(\frac{v_b}{R}\right)}{dt} = J \frac{d^2\left(\frac{x}{R}\right)}{dt^2} = \frac{J}{R} \ddot{x} \quad (3)$$

where,

$F_{rx}$  = exerted force due to ball rotation

$J$  = moment of inertia of solid ball

$\omega_b$  = angular velocity of the ball

$v_b$  = translational velocity of the ball

From (1), (2), and (3) we obtain:

$$F_{rx} = \frac{T_r}{R} = \frac{J}{R^2} \ddot{x} = \frac{2}{5} \frac{mR^2}{R^2} \ddot{x} = \frac{2}{5} m\ddot{x} \quad (4)$$

It is obviously known that  $W_x$  is the force due to mass of the ball ( $m$ ) and gravitational acceleration ( $g$ ) along the  $x$ -axis while neglecting frictions between the ball and the beam, and can be formulated by:

$$\frac{2}{5}m\dot{x} + m\ddot{x} = mg \sin \alpha \quad (5)$$

By dividing both sides by  $m$ , we have:

$$\frac{5}{7}g \sin \alpha = \ddot{x} \quad (6)$$

For  $\alpha$  small,  $\sin \alpha$  is approximated by  $\alpha$ , giving:

$$\frac{5}{7}g\alpha = \ddot{x} \quad (7)$$

Meanwhile, the relationship between the angle servomotor axle and the beam is given by:

$$\alpha = \frac{d}{L}\theta \quad (8)$$

By substituting Equation (8) to (7), we get:

$$\frac{5}{7}g \frac{d}{L}\theta = \ddot{x} \quad (9)$$

Therefore, the transfer function of the ball and beam in  $s$ -domain is given by:

$$\frac{X(s)}{\theta(s)} = \frac{5gd}{7L} \frac{1}{s^2} \quad (10)$$

Constants  $g$ ,  $d$ , and  $L$  are  $9.8 \text{ m/s}^2$ ,  $5 \text{ cm}$ , and  $69 \text{ cm}$ , consecutively. It is very clear from the above transfer function that this system is of second order with double roots on  $s=0$  at the denominator. It also means that this open-loop system has double pole which lie right on the imaginary axis of  $s$ -plane, implying an unstable nature. It makes sense since the ball always keeps rolling down due to the gravity and never rolls back up to reach set points whenever the beam is steadily tilted up. MatLab simulation shows that the step response blows up as the time goes on, as shown in Figure 2.

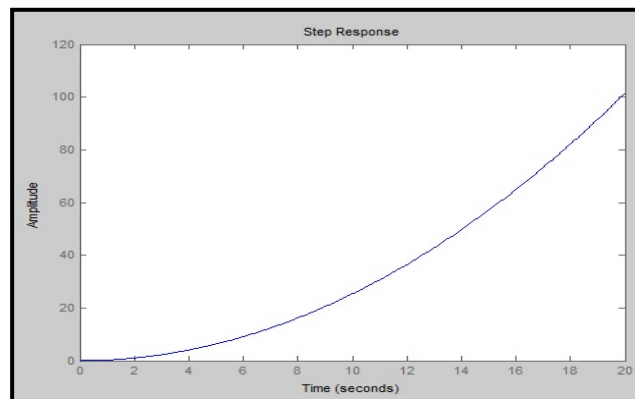


Figure 2. Step response of the open-loop system

### 3. Design of Hardware and Software

The design of the Ball and Beam consists of the closed-loop control system modeling, mechanical and electronic hardware, and microcontroller software. The overview of the system is shown in Figure 3.

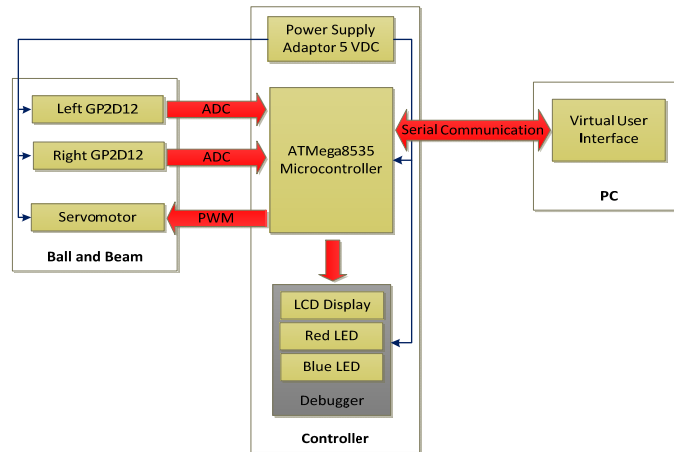


Figure 3. System architecture overview

#### 3.1. Mechanical Design

The designed ball and beam system has a dimension of 90 cm, 15 cm, and 26 cm in length, width, and height, respectively. A billiard ball, having diameter of 5.65 cm and weighing about 180 gram, rolls back and forth by the use of servo-mechanism to reach any desired set points. The mechanical hardware is shown in Figure 4.

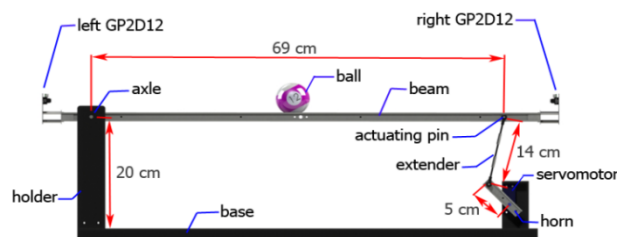


Figure 4. Front view of the mechanical design

Two infrared ranging sensors GP2D12 are placed at both ends of the beam and 5-volt-powered to measure the displacement of the ball. These sensors apply triangulation principle with infrared LED as the emitting element and Position Sensitive Detector as the receiver. An integrated signal processing was pre-embedded in the sensors to produce analog voltages as the output. These sensors are able to detect any objects 10 to 80 cm away. The output voltage of GP2D12 when detecting 10 cm and 80 cm-away objects are 2.6 and 0.4 volts, respectively. The object must be 6 cm in diameter to be properly detected from the distance of 80 cm [11]. In order to acquire valid ball displacement data, the microcontroller sample left and right GP2D12 sensor values with interval of 1 cm. The acquired data are then put into a look-up table, which in turn, build a linear interpolation to determine all possible positions of the ball. The left GP2D12 measures the ball displacement starting from 0 cm to 35 cm, whereas the right GP2D12 measures the ball displacement starting from 35 cm to 60 cm.

The actuator activating the Ball and Beam is a servomotor Turnigy 620MG which is equipped with internal closed feedback inside [12]. A servomotor consists of DC motor, gearbox, variable resistor, and control circuit. The rotational angle of servomotor is determined

by the duty cycle of pulses with period of 20 ms. Horn and extender were assembled at the servomotor axle to adjust its angle and thus tilt the beam up and down. A 5 cm horn was meant to increase the angular velocity of the beam. An R-C (resistor-capacitor) low pass filter which has cut-off frequency  $f_c=1.608$  Hz was implemented by the means of software to stabilize data acquisition of both left and right GP2D12. Suppose sampling period =  $\Delta t$ , then input voltage  $V_{in}(t)$  was sequentially represented as  $x_i, x_{i+1}, \dots, x_n$  and output voltage  $V_{out}(t)$  was sequentially represented as  $y_i, y_{i+1}, \dots, y_n$ , then the discrete low pass filter is given by:

$$y_i = ay_{i-1} + (1-a)x_i \quad (11)$$

Where  $a=RC/(RC+\Delta t)$

In order to monitor the status of the system, a set of debugger was added into system. A red Blue LED indicated the beam status whether "Run" or "Stop". A Red LED and buzzer indicated whether or not a ball was already on the beam. A 16x2 character LCD displayed information regarding to the serial connection status with user interface program as well as the beam status. The heart of the system is the AVR ATmega 8535 manufactured by ATMEL Corporation [13]. This handy chip facilitates internal Analog-Digital Converter, large program and data memory, internal comparator circuit, internal Pulse-Width Modulation, and internal timer. Subroutines of the controller handle time sampling tasks, data acquisition from GP2D12 sensor, data-filtering using low pass filter, data conversion to displacement, PID calculation, servomotor angle adjustment, and connectivity to user interface program via serial communication on PC or laptop for both sending data to be plotted by user interface program and receiving commands from user interface program, as shown in Figure 5.

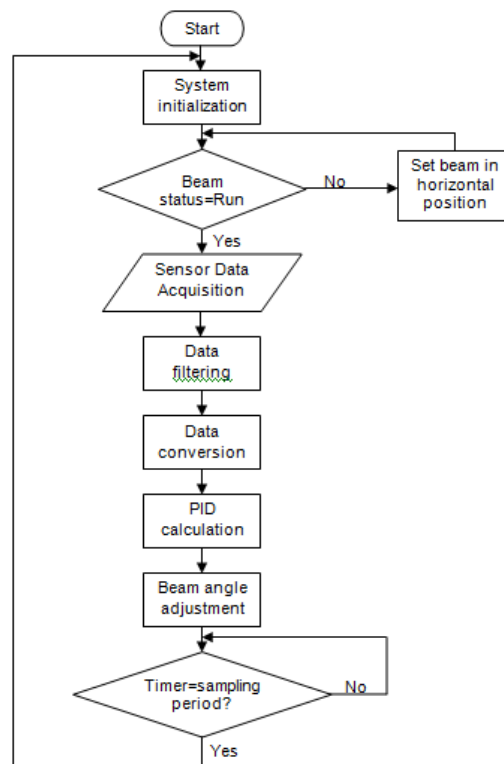


Figure 5. Flow chart of controller subroutines

Figure 6 shows the closed-loop control system consisting of a PID controller, plant (Ball and Beam), and low pass filter.  $C(s)$  is the transfer function of PID controller in s-domain,  $G(s)$  is the transfer function of the plant, i.e. Ball and Beam system, and  $H(s)$  is the transfer function of GP2D12 low pass filter. The PID controller was supposed to make the plant stable by shifting

the open-loop poles, which originally lie on the imaginary axis, to the left-half s-plane. PID calculation was initialized with error calculation, i.e. difference between set points and process variables. The obtained error was the used to calculate manipulated variable ( $MV$ ), which is formulated by:

$$MV = K_p * \text{error} + K_i * \text{sum of error} + K_d (\text{error} - \text{last error}) \quad (12)$$

Then, the value of  $MV$  was used to control servomotor pulse duty cycle through pulse width modulation (PWM). During calculating PID, the integral and derivative sampling period was 50 ms. The transfer function of closed-loop system is now of fourth order and given by:

$$\frac{X(s)}{R(s)} = \frac{0.243K_d s^3 + 2.45(K_d + 0.099K_p)s^2 + 2.45(K_p + 0.099K_i)s + 2.45K_i}{0.478s^4 + 4.83s^3 + 2.45K_d s^2 + 2.45K_p s + 2.45K_i} \quad (13)$$

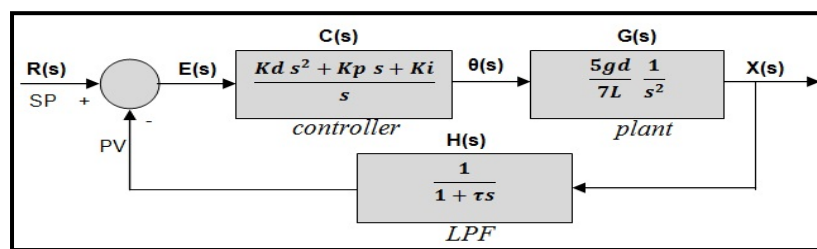


Figure 6. Block diagram of closed-loop system

The realized Ball and Beam system is depicted in Figure 7.



Figure 7. The realized *Ball and Beam*

### 3.2. User Interface

The user interface program was used to adjust PID configuration within closed-loop control system, select the desired set points, display the plot of ball displacement and set points, and calculate the rise time, peak time, maximum overshoot, and settling time, as shown in Figure 8. The application design was accomplished in Visual Studio 2010 with Visual Basic programming language (VB.Net). On the user interface main page, there are three major displays, which are: control panel, plotter display, and info display.

a) Control panel consists of four panels: Beam Status, Plot Data, PID Configuration, and Ball Position. At ball position option, users are allowed to select the desired set points, as shown in Figure 9.

b) Plot display has a grid area to plot the current ball position. The vertical axis represents the displacement (cm) ranging from -10 to 70 cm, while the horizontal axis

represents the time (s). The yellow line indicates the midpoint of the beam. The purple line indicates the ball displacement, while red line represents a marker on which the response starts settling.

c) Info display shows time response characteristics, including: current position, set point, current error, rise time, maximum overshoot, peak time, and settling time.

While communicating with the microcontroller, an updater program handles data-sending task to user interface program as well as data-receiving from user interface program via serial communication with baud rate of 115200 bps. Along the communication, character string packets were transmitted and then translated into instructions or commands. The format of the character string packets received by microcontroller was defined as #[instruction]X. The character “#” was used as the initial mark while the character “X” was used to end the data packet. The instruction itself lies between character “#” and X”. Meanwhile, the microcontroller sent character string packet which no longer needs initial marks but only end marks “/n”.

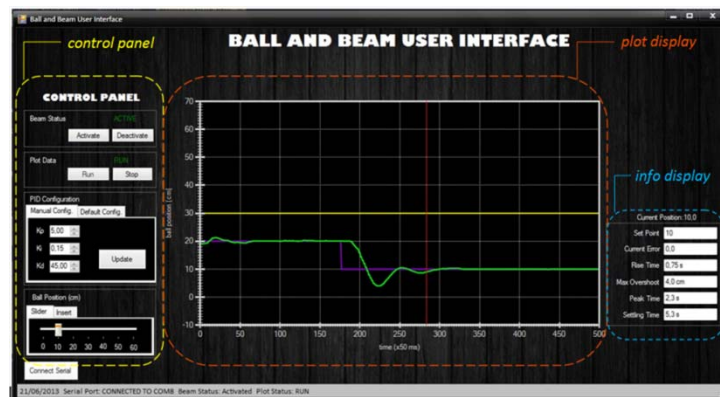


Figure 8. User Interface display

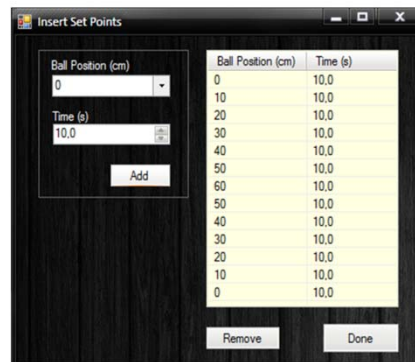


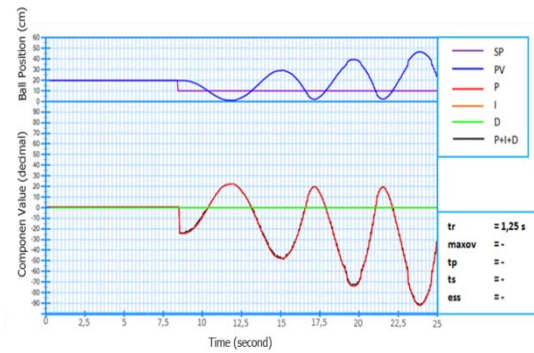
Figure 9. Insert Set Points page

## 4. Results

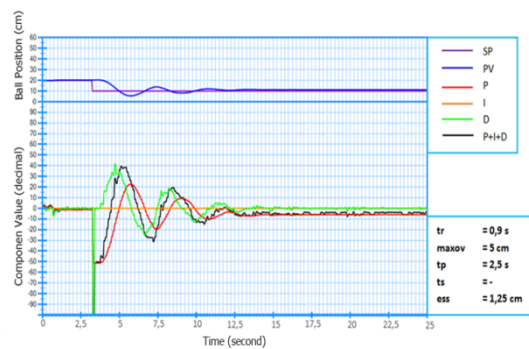
A set of tests was carried out to assess the overall performance of the Ball and Beam training kit.

### 4.1. User Interface

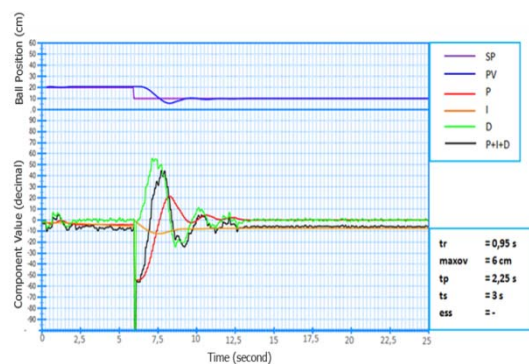
Such tests were done to obtain the optimum  $K_p$ ,  $K_i$ , and  $K_d$  constants with respect to the rise time, maximum overshoot, peak time, and settling time [14-15], as well as to evaluate the performance of the closed-loop system. Testing mechanisms were carried by adjusting set points from 20 cm to 10 cm, as indicated in Figure 10.



(a)



(b)



(c)

Figure 10. System performance using (a)  $K_p=2.5$ ,  $K_i=0$ ,  $K_d=0$ , (b)  $K_p=5$ ,  $K_i=0,1$ ,  $K_d=30$ , (c)  $K_p=5$ ,  $K_i=0,1$ ,  $K_d=45$

The overall performance of the system needs to be evaluated by calculating rise time, maximum overshoot, peak time, and settling time at all set points using specified  $K_p$ ,  $K_i$ , and  $K_d$  configurations. Evaluations were carried by adjusting set points every 15 seconds starting from 0 cm, 10 cm, 20 cm, 30 cm, 40 cm, 50 cm, 60 cm, 50 cm, 40 cm, 30 cm, 20 cm, 10 cm, and finally back to 0 cm. All test configurations were conducted four times and then evaluate the average of rise time, average of maximum peak (Mp), average of peak time, average of settling time, and percentage of successful settling.



#### 4.2. Feedbacks from Students

Seventeen students taking the Control System course were required to fill out a questionnaire as a basic assessment about the use of the microcontroller-based Ball and Beam. The statement in the questionnaire had to be answered on a five-point Likert scale with points 1, 2, 3, 4, and 5 representing “strongly disagree”, “disagree”, “neutral”, “agree”, and “strongly agree”, respectively. After exploring and practicing the Ball and Beam system, they gave their opinion on the following statements: The Ball and Beam helps students to understand closed-loop control system concepts and is easy to perform. Figure 11 shows the questionnaire result of the statement asked to the students. The result indicates that the microcontroller-based Ball and Beam system is an advantageous tool in achieving better understanding of closed-loop control system concepts and very easy to perform. The result of questionnaire was then converted to Rating Value of 94.12% indicating that the students were very satisfied.

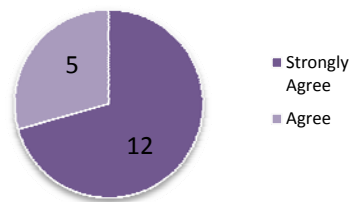


Figure 11. Pie chart of the questionnaire

#### 5. Conclusion

We have developed an ATmega 8535 microcontroller-based Ball and Beam system which consists of infrared ranging sensors, servomotor, PID controller, ball and beam. A virtual user interface, which runs on PC or notebook connected to microcontroller via serial communications, was used to bridge out the system and students. Such user interface was aimed to configure controller coefficients  $K_p$ ,  $K_i$ , and  $K_d$ , to select the set points, to plot the actual ball position, and to display parameter values, namely, actual error, rise time, maximum overshoot, peak time and settling time of the system. The PID constants  $K_p=5$ ,  $K_i=0.2$ ,  $K_d=45$  was chosen to be the default configuration since it has the largest percentage of successful settling, i.e. 93.75%. This Ball and Beam training tool shows conformities with control system theory, in terms of physical modeling, system stability, and closed-loop PID-controlled system. Feedback from the students indicates that the Ball and Beam system is a helpful tool in achieving better understanding of closed-loop control system concepts and very easy to perform.

#### References

- [1] J Sanchez, S Dormido, F Esquembre. The learning of control concepts using interactive tools. *Comput. Appl. Eng. Educ.* 2005; 13(1): 84-98.
- [2] IH Altas, H Aydar, A real-time computer-controlled simulator: For control systems. *Comput. Appl. Eng. Educ.* 2008; 16(2): 115-126.
- [3] S Dormido, S Dormido-Canto, R Dormido, J Sanchez, N Duro. The role of interactivity in control learning. *Int. J. Engng. Ed.* 2005; 21(6): 1122-1133.
- [4] JL Guzman, KJ Astrom, S Dormido, T Hagglund, Y Piguet. Interactive learning modules for PID control. *IEEE Contr. Syst. Mag.* 2008; 28(5): 118-134.
- [5] N Aliane. Spreadsheet-based interactive modules for control education. *Comput. Appl. Eng. Educ.* 2010; 18(1): 166-174.
- [6] B Chen, YC Chou, HH Cheng. Open source Ch control system toolkit and web-based control system design for teaching automatic control of linear time-invariant systems. *Comput. Appl. Eng. Educ.* 2013; 21(1): 95-112.
- [7] SG Nieto, M Martinez, J Salcedo, D Lauri. Practice tool based on open source Scada for experimentation in nonlinear control using the inverted pendulum. *Comput. Appl. Eng. Educ.* 2012; 20(1): 137-148.
- [8] D Ibrahim. Teaching digital control using a low-cost microcontroller- based temperature control kit. *Int. J. Elect. Eng. Educ.* 2003; 40(3): 175-187.

- 
- [9] O Gonzalez, M Rodriguez, A Ayala, J Hernandez, S Rodriguez. Application of PICs and microcontrollers in the measurement and control of parameters in industry. *Int. J. Elect. Eng. Educ.* 2004; 41(3): 265-274.
- [10] D Ibrahim, T Alshamleh. An undergraduate fuzzy logic control lab using a line following robot. *Comput. Appl. Eng. Educ.* 2011; 19(4): 639-646.
- [11] Sharp Corp., GP2D12 Optoelectronic Device Datasheet, 2005, [http://www.sharpsma.com/webfm\\_send/1203](http://www.sharpsma.com/webfm_send/1203) [accessed on September 26, 2014].
- [12] Hobby King, Turnigy 620DMG+HS High Torque Digital Servo (MG) Datasheet, "[http://www.hobbyking.com/hobbyking/store/\\_\\_9441\\_\\_Turnigy\\_620DMG\\_HS\\_High\\_Torque\\_Digital\\_Servo\\_MG\\_10\\_6kg\\_13sec\\_52g.html](http://www.hobbyking.com/hobbyking/store/__9441__Turnigy_620DMG_HS_High_Torque_Digital_Servo_MG_10_6kg_13sec_52g.html)" [accessed on September 15, 2014].
- [13] Atmel Corporation, AVR ATmega 8535 Datasheet, 2006, [www.atmel.com](http://www.atmel.com) [accessed on September 12, 2014]
- [14] CL Phillips, RD Harbor. Feedback Control Systems. Second Edition. Englewood Cliffs, New Jersey: Prentice Hall. 1991: 124.
- [15] NS Nise. Control Systems Engineering. Fourth Edition. New Jersey: John Wiley & Sons. 2004: 193.