

Development of An Experimental Setup for the Altitude Control of A Ball in A Pipe

Şeyma AKYÜREK^{1,a}, Gizem Sezin ÖZDEN^{1,b}, Coşku KASNAKOĞLU^{1,c}

¹TOBB University of Economics and Technology, Ankara, Turkey

^aseymaakyurek@gmail.com, ^bsezin.ozden@gmail.com, ^ckasnakoglu@gmail.com

Key Words: Altitude control, PID controller, Arduino, ultrasonic range sensor, software in the loop.

Abstract. In this paper we consider the development of a ball-and-pipe system controlled by an Arduino-based PID controller designed and tuned numerically in MATLAB/Simulink environment. The controller achieves small overshoot and fast settling time. The system is intended to be used as a laboratory experiment in control systems theory courses.

Introduction

Laboratory experiments play an important role in engineering education [1] and in control systems theory courses in particular [2]. Examples include DC motor experiments [3], coupled tank apparatuses [4] and magnetic suspension control systems [5]. Most of these experiments employ some sort of PID control, which is the most widely used control method in both educational and industrial control systems. This approach is often combined with logic, sequential functions, selectors, and simple function blocks to build complicated automation systems. PID controller algorithm involves three separate constant parameters: proportional (P), integral (I) and derivative (D); the coefficient P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors [6].

Many current PID control systems make use of the Arduino, a single-board microcontroller, which makes it easy to code the controller, obtain inputs and provide outputs. The Arduino integrated development environment (IDE) is a cross-platform application written in Java, and derives from the IDE for the Processing programming language and the wiring projects [7].

In this study we build a control system experimental setup to control the altitude of a ball inside a pipe by using air blown from below. The PID controller is implemented using an Arduino Uno microcontroller board which obtains distance readings from an ultrasonic sensor and outputs the necessary commands to drive the motor for a blower providing air flow from the bottom of the pipe.

System Overview

Altitude position control system consists of Arduino Uno, ultrasonic sensor, blower fan and a ball. In the present work, DC air blower, HC-SR04 ultrasonic ranging module, Arduino Uno, 50cm Plexiglas pipe and a foam ball are integrated to build a height control system.

Ultrasonic ranging module provides 2cm-400cm non-contact measurement function and the ranging accuracy can reach 3mm. The module includes ultrasonic transmitters, receiver and control circuit [8]. The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button [9]. For air flow DC Air Blower DC-P-R-3 is chosen with the following properties: current is about 1Amp at 12VDC, rated airflow is 16CFM, rated power is 10W, outlet diameter is 33mm, fan speed is 3000-3500 rpm [10]. A transparent Plexiglas pipe is used to be able to see the ball's current position.

Blower and ranging sensor are placed next to each other to build a more compact structure. The blower and ultrasonic sensor are positioned in a wooden box. The top portion of the wooden box is drilled to make Plexiglas pipe fit.

Plexiglas pipe's length is 50 cm, diameter is 8 cm, which is close to the ball's diameter. These two measurements have to match since otherwise the ball will not go up and shake. If there are big

gaps, vibration occurs because of the forces which are created by air the blower provides and hits the pipe's surface. Therefore, the ball's diameter is chosen as 7.5 cm.

For adjusting the air flow rate, current control was chosen since it was determined to result in a faster response. The operation range for the blower is 0.6 A-1.0 Amp, which is discretized into small increments for the digital control of the blower speed through the Arduino based controller. A photo of the system is given in Figure 2.

Software

In this section the software used in the design is explained. The distance of the ball is measured by the ultrasonic range sensor HCSR-04. A three-point median filter is used to eliminate spikes in the measurements. A set point altitude is input to the software and the difference between the set point and current altitude is the error, which is the input of the PID controller. The output of the PID controller is the current command, which is sent to the blower driver so as to control the air flow speed and hence adjust the ball's altitude. The current control driving the blower is shown in Figure 3. The reason for the current control circuit design is to be able to achieve a quick response and to operate in the narrow range required for the setup at hand. The current control circuit utilizes a TIP 120 Darlington transistor since the Arduino itself cannot provide the high current required by the blower motor.



Fig. 1. A photo of the system

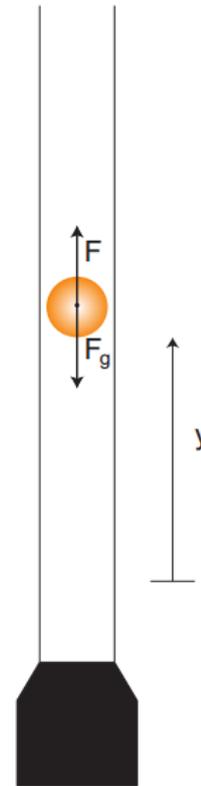


Fig. 2. Schematic illustration of the system

The overall block diagram of the controller can be seen in Figure 4. The next section describes the design and tuning of the PID controller.

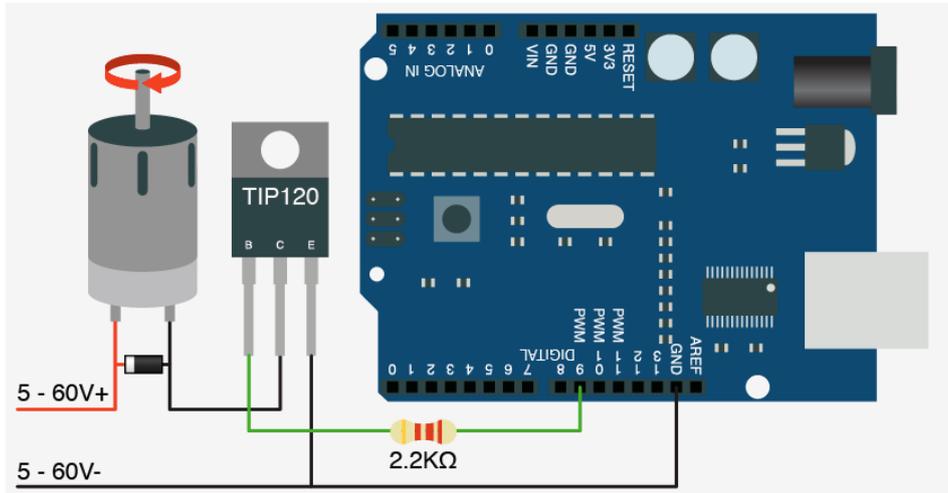


Fig. 3. Arduino circuit connections

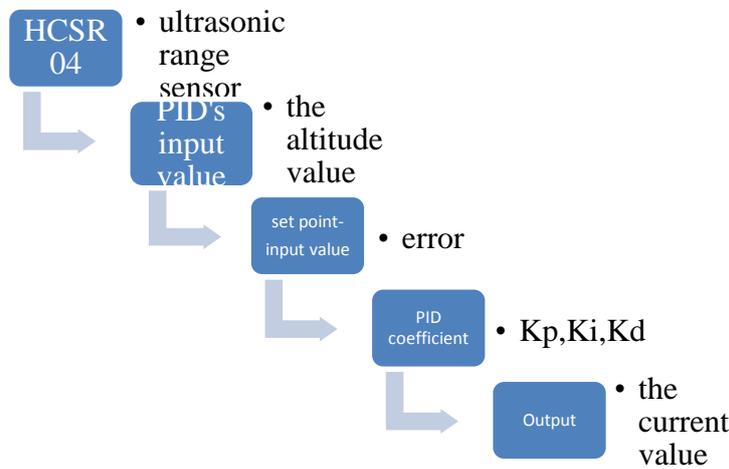


Fig. 4. Block diagram of the software system

Controller Design

To design the controller firstly the equations of motion of the system need to be calculated. Based on the mass of the ball m , the gravity g and the force F which is provided by blower the following can be written from Newton's Law:

$$F - mg = m \frac{dy^2}{dt^2}$$

where the frictional forces are assumed to be small and therefore ignored. The system equation as such is affine but since mg is constant we can shift the input as $\bar{F} = F - mg$ to obtain a linear equation whose Laplace transform and transfer function can be obtained as:

$$\bar{F}(s) = ms^2Y(s) \Rightarrow \frac{Y(s)}{\bar{F}(s)} = \frac{1}{ms^2} .$$

We can assume $m = 1$ without loss of generality since we can compensate for this assumption by scaling the PID controller coefficients accordingly. The controller coefficients are calculated by the help of automated tuning procedures available in MATLAB/Simulink, the model for which is

shown in Figure 5. The results of the tuning procedure are shown in Figure 6 and the closed loop step response is shown in Figure 7. It can be observed the system settles in about 1 second with little overshoot, which is satisfactory for the problem at hand. This controller was then implemented in Arduino code and verified on the actual experimental setup; the corresponding figures could not be included here due to space limitations but they were consistent with Figure 7.

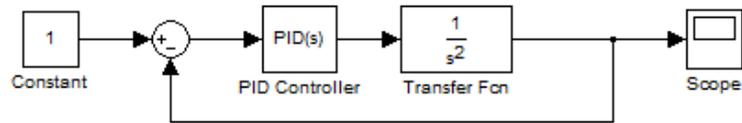


Fig. 5. PID controller system in MATLAB/Simulink

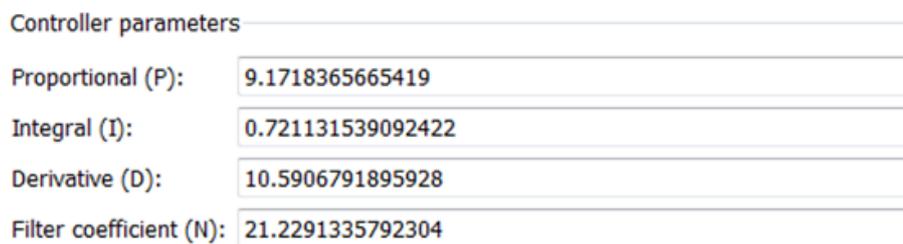


Fig. 6. Tuned PID parameters in MATLAB/Simulink

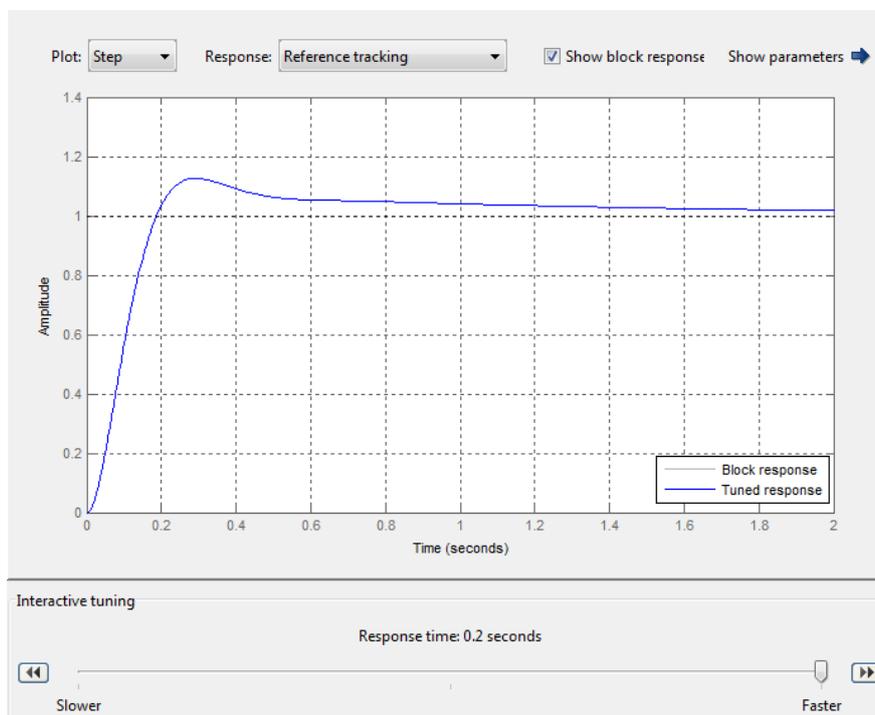


Fig. 7. Step response of the PID controller designed

Conclusions and Future Works

In this paper an experimental setup was developed to keep a ball in a pipe at a desired altitude. A PID controller was designed and tuned using MATLAB/Simulink with small overshoot and fast settling time. The controller was implemented using an Arduino Uno board to control the air flow

to the system via adjusting the current to the blower motor. The experimental setup could provide useful for control education.

Future works include developing alternative controllers for the ball-and-pipe system and constructing experimental setups for HVAC (heating, ventilating, and air conditioning) systems [11].

References

- [1] Feisel, L. D. and Rosa, A. J. ,“The Role of the Laboratory in Undergraduate Engineering Education,” *Journal of Engineering Education*, 94: 121–130. DOI: 10.1002/j.2168-9830.2005.tb00833.x , January 2005
- [2] Horáček, Petr. "Laboratory experiments for control theory courses: A survey." *Annual Reviews in Control* 24 : 151-162, 2000
- [3] Yeung, Kin, and Jie Huang. "Development of a remote-access laboratory: a dc motor control experiment." *Computers in Industry* 52.3 : 305-311, 2003
- [4] Ko, Chi Chung, et al. "Development of a web-based laboratory for control experiments on a coupled tank apparatus." *Education, IEEE Transactions on* 44.1 : 76-86, 2001
- [5] Oliveira, Vilma A., Eduardo F. Costa, and Jerson B. Vargas. "Digital implementation of a magnetic suspension control system for laboratory experiments." *Education, IEEE Transactions on* 42.4 : 315-322, 1999
- [6] Åström, Karl J., and Tore Hägglund, *PID controllers: theory, design, and tuning*, Instrument Society of America, Research Triangle Park, NC , 1995.
- [7] Banzi, Massimo, *O'Reilly Media, Inc. Getting Started with Arduino*, 2009.
- [8] Elec Freaks, Ultrasonic Ranging Module HC - SR04 Datasheet, Available: http://www.elec Freaks.com/store/download/product/Sensor/HC-SR04/HC-SR04_Ultrasonic_Module_User_Guide.pdf. Retrieved 2014-12-14.
- [9] Arduino Project, Arduino Uno, Available: <http://arduino.cc/en/Main/arduinoBoardUno>. Retrieved 2014-12-14.
- [10] Sparkfun Electronics, Blower - Squirrel Cage (12V), Available:<https://www.sparkfun.com/products/11270>. Retrieved 2014-12-14.
- [11] *Ventilation and Infiltration Chapter, Fundamentals Volume of the ASHRAE Handbook*, ASHRAE, Inc., Atlanta, GA, 2005.