

DEVELOPMENT OF AN INTERACTIVE HEIGHT MEASUREMENT DEVICE USING AN ARDUINO CONTROLLER – AN INDEPENDENT STUDY PROJECT

Tristan Cowan, Qing Li
Mechanical Engineering Department,
University of New Haven

Abstract

Arduino is a compact, inexpensive microcontroller with open-source software and hardware available on the internet. The technical features, low cost and small size of the Arduino controller have made it an ideal tool for mechatronics education. In this Independent Study Project course, the student is required to design and develop a mechatronic device - an interactive height measure using an Arduino board. This measurement device can control the speed of a blower to levitate a ball to match a user's height. The construction of this mechatronic system incorporates aspects of mechanical, electrical, microprocessor, and control engineering, and thus demands a challenging integrated multidisciplinary design and development effort for the student. The immediate visual response to the presence and height of an object invites the user's interaction and inquiry. This engaging mechatronic device can also be used as an attractive demonstration in an engineering open house to promote public interest in mechatronics and automation.

Introduction

An Independent Study Project (ISP) course allows students to extend their education beyond the courses that are offered in the Tagliatela College of Engineering at the University of New Haven [1]. Normally, ISP courses are designed to allow students to work one-on-one with professors to explore a topic that is of interest but not taught in a class in the college. An ISP can be based on academic research or industrial interests. The most important aspect of the ISP course is to create a unique learning experience that does not involve material that is duplicated by any course offered at the university.

Therefore, ISPs require advance special instructional material, planning and faculty approval before a student may register for academic units.

The senior student registered in this ISP course double majors in both mechanical and electrical engineering. He needed a special learning experience that integrated and synergized the knowledge acquired from these two engineering programs. Since "mechatronics is the synergistic combination of mechanical engineering, electronic engineering, control engineering, and computer engineering, all integrated through the design process" by definition [2], a mechatronics based project course would best suit this student's specific learning needs. The professor supervised this ISP and proposed the concept of the interactive height measurement device. The intention of this proposal is twofold: 1) to provide the student an opportunity to fully explore every aspect of a mechatronic system and then integrate them together to build a functioning automatic device, and 2) to develop an interesting educational tool that explains the engineering concepts of *mechatronics*, *automation*, and *sensing* to the students and the public in an interactive engaging way.

Arduino is a compact, inexpensive, open-source electronics prototyping platform built around an Atmel AVR microcontroller [3]. A single board system programmed via USB connection to a host computer, Arduino has regulated and unregulated direct current (DC) power, digital inputs and outputs, and analog inputs. In recent years Arduino has become part of a larger trend toward open source material fostered by a diverse mix of private tinkerers and companies selling electronics kits [3]. The

broad interest and commercial viability of Arduino make it easy for academics to focus on the development of instructional materials, not on the design, fabrication, and support in the traditional way. The technical features, low cost and small size of the Arduino controller have also made it an ideal tool for teaching microprocessor and mechatronics topics. In this project, an Arduino Uno board is adopted as the central processing unit to read a sensor signal, generate a control output, and actuate a motor for the height measurement device.

As the project outcome, the student has successfully designed and developed the interactive height measure. The device has been displayed in several Accepted Student Days and Engineering Open Houses. The instantaneous automatic floating of the ball in response to the presence of an object has caught students' curiosity and inspired their learning motivation. This ISP course has provided a valuable learning experience, requiring the student to extend and adapt previously studied concepts to a real world setting and to consider practicalities of design and implementation which are implicit in engineering curricula, but which are frequently difficult to experience within a classroom.

Product Design

The illustrative CAD model of the interactive height measurement device is shown in Figure 1. It consists of a base holding the blower and electrical items, a clear polyvinyl chloride (PVC) pipe mounted on top of the blower with a light-weight foam ball inside, and two infrared sensors, one positioned to the side of the device to measure a user's height, and the other positioned at the top of the pipe to measure the height of the ball and provide a feedback mechanism. Building this device involves mechanical structure design, electrical circuit design, infrared (IR) sensor reading, Arduino programming and proportional-integral-derivative (PID) tuning. The components used for this product are listed in

Appendix A. The details of each design Procedure are described below.

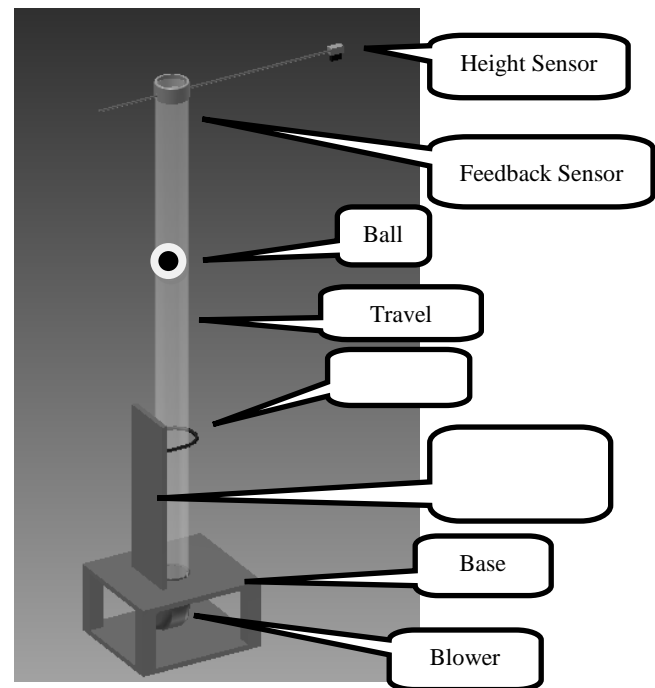


Figure 1: CAD model of device

Mechanical Structure Design

Several design challenges needed to be overcome in creating this structure. A method of holding the pipe steady was required, as was a method of holding the height measurement sensor at a point far enough away from the base to allow a person to stand comfortably underneath it. Moreover, the entire device needed to be easily assembled and disassembled for transportation.

To address the first concern, a vertical plank was affixed to the top of the base and the pipe was held to it with a u-bolt, as shown in Figure 1. This keeps the pipe vertical with a minimum of play while also allowing for easy disassembly. To mount the sensors, a length of threaded rod was inserted through the top of the pipe and held in place with a PVC end-cap, modified to have an open top and a secure fit over the rod (see Figure 2). Two aluminum blocks were machined to thread onto the rod and provide mounting points for the sensors (see Figure 3). To simplify control signal

computation and programming, a single rod is used to mount both sensors so that they are placed on the same plane.

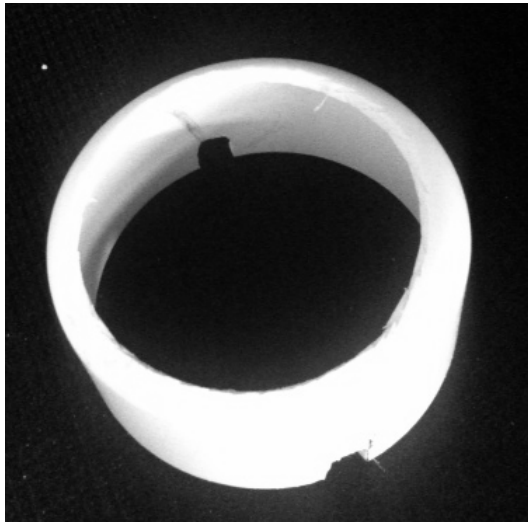


Figure 2: Modified end-cap.

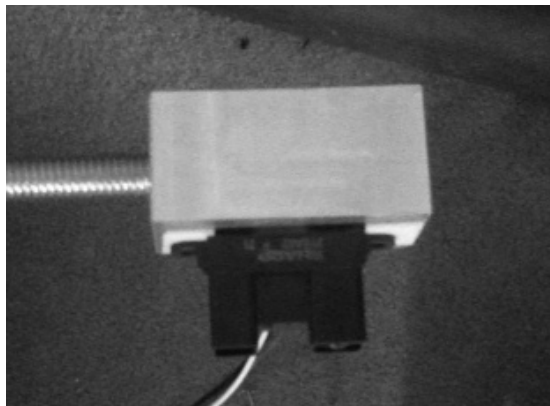


Figure 3: Sensor on mounting block.

Window screening is secured over the intake and outtake of the blower as a safety measure. This can prevent the ball from becoming lodged in the blower exhaust. The base of the measurement device is painted black and the ball white to make the ball easier for the sensor to pick up. The modular concept has been applied throughout the design so the finished device can be disassembled to the pipe, base, and sensor rod components, greatly simplifying transportation. The price paid for this convenience is the necessity of performing realignment on each assembly to ensure that sensors are positioned such that the feedback sensor beam is clear of the pipe.

Electrical Circuit Design

Figure 4 illustrates the electrical schematic diagrams of the system. A 12V, 30A power supply was selected to provide the blower (a Jabsco 35770-0092 12 V Flexmount blower) with power. Control of the motor speed was achieved by pulse width modulation (PWM) using a MegaMoto motor controller with an Arduino Uno microcontroller. The MegaMoto controller is able to handle the high current drawn by the blower (up to 12A continuous). The power is fused at 15A as per the blower specifications, providing an extra measure of safety in case of overcurrent. Pins 11, 8, and 3 of the Arduino Uno are connected to the MegaMoto controller to control the motor (see Figure 4). The input and set-point sensor signals are received on analog pins A0 and A2. IR sensors were selected over ultrasonic sensors due to their narrow beam width. The sensors used are the Sharp GP2Y0A02YK distance sensors, which have a detection distance of 20cm to 150cm, wide enough to detect people of typical height range. These sensors have relatively low sensitivity to differences in color and lighting, and can be connected directly to a microcontroller without the need for an external circuit. The IR sensor power was supplied directly from the microcontroller 5V supply, and the microcontroller itself was provided with a separate 12V supply in order to separate high and low current functions. The electrical circuit box is shown in Figure 5.

Programming

The programming code for the device is shown in Appendix B. In addition to the built-in math library, two freely available libraries for Arduino are used: the Arduino PID library created by Brett Beauregard [4] which provides a flexible and reliable library of functions for PID control, and the Smooth Analog Input library by Richard Lee which creates a SmoothAnalogInput object averaging 32 analog reads of the specified input [5].

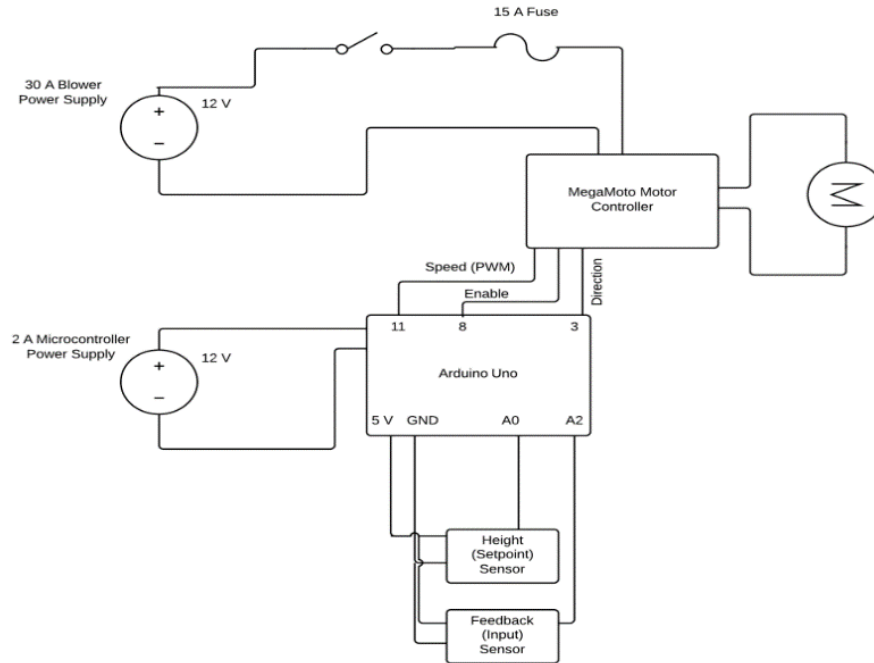


Figure 4: Electrical Schematic.

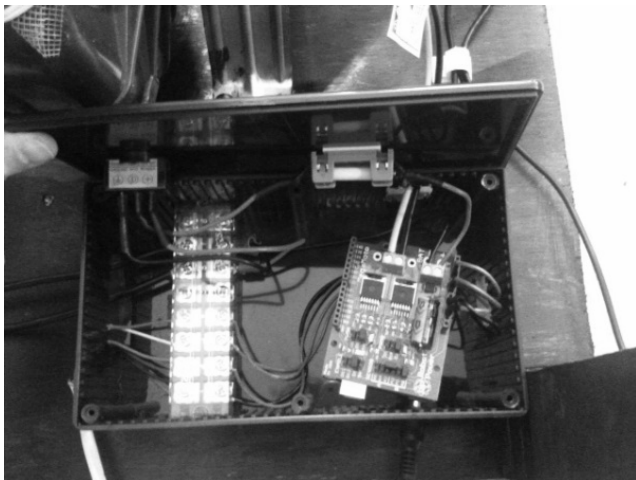


Figure 5: Circuit box.

The MegaMoto controller is configured in H-bridge mode, with the enable pin set to pin 8, the speed control set to pin 11, and the direction control set to pin 3. Smoothed analog input objects are created for both IR sensors to reduce sensor jitter and produce smoother control. The PID control algorithm is set up and the inputs initialized. During the “loop” portion of the program, the motor is enabled and the

direction set and the input and set-point sensors are read. In this portion of the code, a correction function is applied to the set-point sensor to compensate for the differences in sensor response caused by reflection from the walls of the pipe. The PID control output is limited to 0 to 90 (of a full range of 0 to 255), as the blower at full speed is powerful enough to eject the ball from the pipe. A range of 0 to 90 permits a rapid PID response while preventing a violent collision of the ball and feedback sensor. The PID control output is then calculated and output as a PWM signal on the motor speed pin. If a serial port is available, set-point and input sensor data are transmitted via serial to facilitate troubleshooting.

PID Tuning

For the purposes of tuning the PID control, the set-point was manually set to a fixed value within the program in order to eliminate unwanted external disturbances. Ziegler-Nichols tuning [6] was then followed to establish a set of starting values. In the Ziegler-Nichols tuning method, the integral and

derivative terms (K_i and K_d) are first reduced to zero and the proportional term (K_p) is increased until the system oscillates constantly at a steady amplitude. The value of K_p which produces this oscillation is the ultimate gain for the PID control, K_u , and the period of oscillation is the ultimate period, T_u . These values may be used to estimate suitable values of K_p , K_i , and K_d for the system. Table 1 shows some common methods of calculating PID constants using these values.

Control Type	K_p	K_i	K_d
P	$0.5K_u$	-	-
PI	$0.45K_u$	$1.2K_p/T_u$	-
PD	$0.8K_u$	-	$K_p T_u/8$
Classic PID	$0.6K_u$	$2K_p/T_u$	$K_p T_u/8$
Pessen Integral	$0.7K_u$	$2.5K_p/T_u$	$0.15K_p T_u$
Some overshoot	$0.33K_u$	$2K_p/T_u$	$K_p T_u/3$
No overshoot	$0.2K_u$	$2K_p/T_u$	$K_p T_u/3$

Table 1: PID constants calculated by the Ziegler-Nichols method.

The “no overshoot” method was chosen as a starting point for this system, as a smooth response was desired and the aggressiveness of the blower favored a low K_p value. K_u was found to be equal to 1, with T_u of 2 seconds, yielding initial K_p of 0.2, K_i of 0.2, and K_d of 0.13. This set of constants yielded a system with a quick settling time and apparently accurate settling level, but a still undesirable level of overshoot and oscillation.

Sensor Reading Troubleshooting

In the course of adjusting the PID constants to attempt to eliminate the oscillation, it was discovered that there existed a discrepancy between sensor readings for taller heights. For example, for a person standing at 5 feet 5 inches, the set-point sensor might read a value of 255. However, the input sensor would read a corresponding 255 value with the ball floating at 5 feet only. This discrepancy was much less pronounced for shorter heights. As a first step in troubleshooting, sensors were switched, and realignment was performed. These checks

verified that both sensors were performing to specifications and that the pipe walls were not impinging on the input sensor beam. Input and set-point sensor values sent over serial communications verified that the PID control was performing as desired and correctly manipulating the ball to match the input and set-point sensor readings.

Given that these checks passed, it was suspected that the sensor beam was reflecting off the ball and rebounding off the pipe walls. To combat this, a logarithmic compensation algorithm was implemented to adjust for the difference. As the use of this algorithm necessarily resulted in a more aggressive response at higher values, the PID control was readjusted to be much less reactive, with final values of $K_p = 0.05$, $K_i = 0.03$, and $K_d = 0.05$.

Final Product

As illustrated in Figures 6, 7 and 8, the final product can smoothly and accurately match the height of a user. When the subject height varies, the IR sensor will pick up the new distance and feed the signal to the blower’s PID controller for speed adjustment. More interesting operations can be found in this link: <http://www.youtube.com/channel/UCYXd95GpOvOersgMTB4D8BA>

However, there are some caveats of the device. The height sensor is extremely sensitive to user positioning and attitude – if the user’s head is tilted or out of line with the sensor, an inaccurate height measurement will result and the device will match the height of a shoulder, for example. The varying reflectivity of hair can also cause aberrations in sensor performance, and optimal results are obtained while wearing a hat to provide a larger target and more uniform reflectivity. During initial assembly of the device, it is important to ensure that sensors are aligned such that the feedback sensor does not experience interference from the walls of the pipe. However, provided these requirements are met, the result is an engaging and educational interactive device.

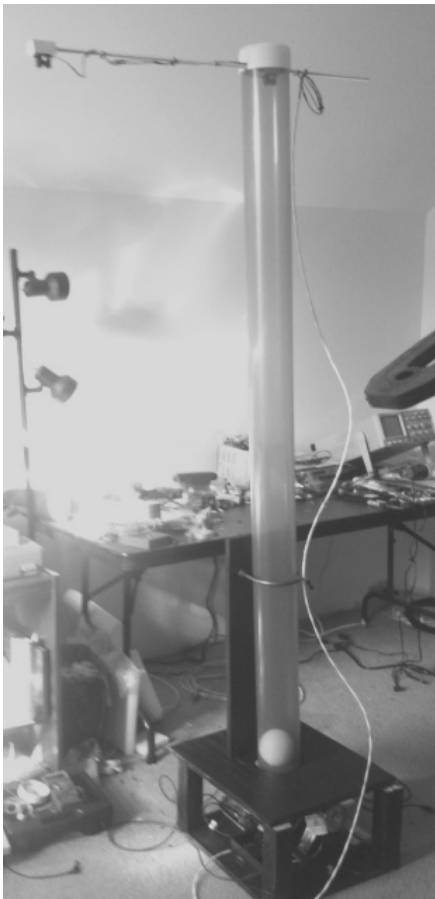


Figure 6: The setup of the measurement device.

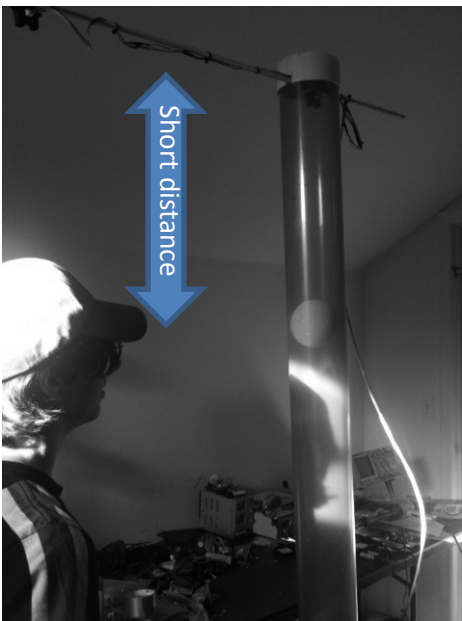


Figure 7: Height matching for standing subject.

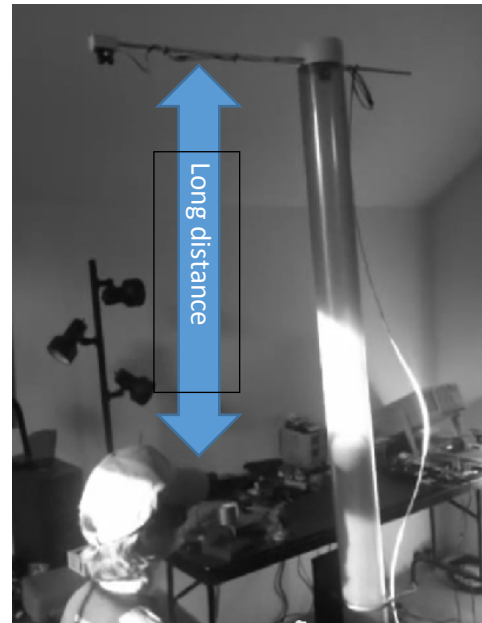


Figure 8: Height matching for kneeling subject.

Educational Value

An Independent Study Project course such as this provides a unique opportunity to enhance student learning by offering a relatively free-form environment in which to develop and test designs. This permits the student to directly experience and understand the design process by offering the opportunity to apply classroom knowledge to a real device. The student is required to suit the design to match the realities of what is practical, and may encounter problems of implementation not typically covered in a classroom setting.

In this project, learning above and beyond the conceptual classroom level was achieved in a number of areas.

- It was initially anticipated that the structural design of this project would present fewer issues. Virtually no loading is present in the device, and there are no moving parts where fatigue or friction could occur. However, the necessity of creating a device that could be easily disassembled and reassembled for transportation purposes imposed a set of “real world” design constraints.

Modular design was the solution to this problem.

- A similar set of “real world” issues were present in the selection of sensors and the tuning of the control system. Instrumentation and controls had previously been studied in the classroom, however, applying this knowledge to a real system introduced a number of environmental variables and practical issues which had to be taken into account. The student gained first-hand experience with the reality of sensor jitter and the necessity of applying filtering for a system (like the smoothing functions from the Smooth Analog Input library) which responds to instantaneous measurement. Working with a sensor in an environment which introduced noise (the sensor inside the pipe, which was subject to interference from the pipe walls) required the student to observe the problem, eliminate potential sources of error, and apply a creative approach to solve the issue programmatically by introducing a compensation algorithm. Likewise, while PID control is frequently studied as part of an elementary controls course, the realities of its application in a particular system require the student to develop skills in systematic observation and troubleshooting.
- The selection of an appropriate motor and the electrical wiring to power and control it also introduced practical problems of implementation for the student to overcome. Although blowers of adequate power were easily found, the consideration of factors such as size, cost, power requirements, and suitability for speed control and continuous duty added an additional layer of complexity to the design. Likewise, although the electrical schematic of the system is relatively simple, the physical circuit box and wiring required the student to consider space requirements, safety, and organization of

valuable engineering concepts which are often divorced from theoretical work in the classroom.

In summation, this project constituted a valuable learning experience for the student, requiring them to extend and adapt previously studied concepts to a real world setting and to consider practicalities of design and implementation which are implicit in engineering curricula, but which are frequently difficult to experience within a classroom.

Conclusion

A hands-on project-based independent study is presented in this paper. The important educational aspect of this ISP course is to provide a unique learning opportunity that enables the student to explore the design and development of a mechatronics device. Most importantly, the *concurrent engineering* design concept that is the core in mechatronics system design [6] is emphasized and explored in this study. In the traditional classroom-based instruction, the student is already given the opportunity to acquire the knowledge in mechanical and electrical engineering courses. While in this project, he was provided the learning experience to understand electro-mechanical integration. That is, the mechanical structure design, electrical circuit creation, sensor reading, PID control, and programming should be considered simultaneously in a mechatronics system design. Furthermore, all these components should be integrated concurrently during the implementation state in order to achieve the optimized performance for the mechatronics product.

The successfully developed project has reinforced the teaching material that was covered in the previous courses. The student was able to retain and enhance the knowledge gained through this reinforcement. In addition, developing an entire mechatronics device also extended the learning experience beyond the courses that are offered at the university.

References

1. University of Newhaven, Undergraduate Catalog, 2012-2014.
2. K. Craig, "Mechatronics in University and Professional Education: Is There Anything Really New Here?" IEEE Education Special Issue of the Journal of Robotics and Automation, vol. 8, pp. 12-19, 2001.
3. Arduino, "Language Reference," <http://arduino.cc/en/Reference/HomePage>, (accessed April 2014).
4. B. Beauregard, "Arduino PID Library," <http://playground.arduino.cc/Code/PIDLibrary>, (accessed April 2014).
5. R. Lee, "Smooth Analog Input Class for Arduino," <http://rl337.org/2012/01/11/smooth-analog-input-class-for-arduino/>, (accessed April, 2014).
6. Wikipedia. "Ziegler-Nichols Method," http://en.wikipedia.org/wiki/Ziegler%E2%80%93Nichols_method, (accessed April 2014).
7. Q. Li, W. J. Zhang, and L. Chen, "Design for Control - A Concurrent Engineering Approach for Mechatronic Systems Design," IEEE/ASME Transaction on Mechatronics, vol. 6(2), pp.161-169, 2001.

Biographical Information

Tristan Cowan graduated from the University of New Haven in May 2014 with degrees in mechanical and electrical engineering. He is currently employed at APS Technology, Inc. in Wallingford, CT, USA.

Dr. Li is an Associate Professor in the Mechanical Engineering Department at University of New Haven. She has unique cross disciplinary educational backgrounds in Mechatronics Engineering (Control & Robotics) and Educational Psychology (Educational Program Measurement & Assessment). Her research interests include robotics, artificial intelligence and educational measurement. She has taught a broad range of courses in applied statistics, general engineering, kinematics and dynamics, systems and control, robotics and mechatronics at universities in Australia, Singapore, and US.

Appendix A: Component List

Item	Qty	Price	Total	Notes
4" Clear PVC Pipe, 7 ft	1	\$ 121.66	\$ 121.66	http://flexpvc.com/cart/agora.cgi?product=PVC-CLEAR-PIPE-UV-Thinwall
PVC endcap	1	\$ 10.48	\$ 10.48	http://flexpvc.com/cart/agora.cgi?product=PVC-Caps-Flat
Cushioned pipe clamp	1	\$ 12.64	\$ 12.64	http://www.grainger.com/Grainger/CADDY-Pipe-Clamp-1RVD3
3/8-16 Threaded rod, 3 ft	1	\$ 6.33	\$ 6.33	http://www.grainger.com/Grainger/Threaded-Rod-4FGK5?Pid=search
Machined block	1	\$ -	\$ -	Use shop scrap
IR sensor	2	\$ 15.95	\$ 31.90	http://www.adafruit.com/products/1031
Nerf Blaster Ball	1	\$ 6.99	\$ 6.99	http://www.toysrus.com/product/index.jsp?productId=2326489
U-bolt	1	\$ 6.98	\$ 6.98	http://www.grainger.com/Grainger/CADDY-U-Bolt-With-Nuts-1CWJ5?Pid=search
3/4" 2'x4' Plywood Project Pack	1	\$ 23.76	\$ 23.76	http://www.homedepot.com/p/3-4-in-x-2-ft-x-4-ft-Pine-Plywood-Project-Panel-2-Pack-1502108/203444162#.UjsyTD_hc2Q
1 1/2" corner braces (4 pack)	1	\$ 2.67	\$ 2.67	http://www.homedepot.com/p/Everbilt-1-1-2-in-Zinc-Plated-Corner-Braces-4-Pack-15304/202033892#.Ujsy_z_hc2Q
Noctua NF-A15 Fan	1	\$ 18.79	\$ 18.79	http://www.amazon.com/Premium-Quality-Cooling-NF-A15-PWM/dp/B00AED7XFI/ref=sr_1_1?ie=UTF8&qid=1379783831&sr=8-1&keywords=noctua+nf-a15
Motor controller	1	\$ 19.95	\$ 19.95	http://www.adafruit.com/products/1438#Description
Microcontroller	1	\$ 24.95	\$ 24.95	https://www.sparkfun.com/products/11021
Hammond Project Boz 7.5"x4.3"x2.2"	1	\$ 9.91	\$ 9.91	http://www.amazon.com/Hammond-1591ESBK-ABS-Project-Black/dp/B0002BSRIO/ref=sr_1_1?ie=UTF8&qid=1379785481&sr=8-1&keywords=electronics+enclosure
12 VDC 2A Power Supply	1	\$ 9.88	\$ 9.88	http://www.amazon.com/Switching-Power-Supply-Adapter-110V/dp/B006NTNGN0/ref=cm_cr_pr_product_top
Total			\$306.89	Does not include tax and shipping

Appendix B: Arduino Program

```

/*****
*Height Measurement Device with PID Control
*Tristan Cowan
*Mechatronics Concentration
*Mechanical Engineering Department
*University of New Haven
*****/

//Motor variables
int EnablePin = 8;
int PWMPin = 11;
int PWMPin2 = 3;

//PID variables
double Setpoint, Input, Output, SetpointHolder;
float Kp = 0.05;
float Ki = 0.03;
float Kd = 0.05;

#include <PID_v1.h>
#include <SmoothAnalogInput.h>
#include <math.h>

//Sensor inputs
SmoothAnalogInput inputIR;
SmoothAnalogInput setpointIR;

//Timing
unsigned long serialTime;

void setup()
{

```

```

//Initialize digital pins
pinMode(EnablePin, OUTPUT);
pinMode(PWMPin, OUTPUT);
pinMode(PWMPin2, OUTPUT);

//Initialize analog pins
inputIR.attach(A2);
setpointIR.attach(A0);

//Initialize PID variables
Input = analogRead(A2);
Setpoint = analogRead(A0);

//Start PID
myPID.SetMode(AUTOMATIC);

//Start serial communications
serialTime = 0;
Serial.begin(9600);
}

void loop()
{
//Start timing
unsigned long now = millis();

//Enable motor and set direction
digitalWrite(EnablePin, HIGH);
analogWrite(PWMPin2, 0);
//Get setpoint and feedback from IR sensors
Input = inputIR.read();
SetpointHolder = setpointIR.read();
if(SetpointHolder <= 50){
  Setpoint = SetpointHolder;
}
else{
  Setpoint = SetpointHolder +
    14*log(SetpointHolder);
}

//Limit output range and send PID to motor
myPID.SetOutputLimits(0, 90);
myPID.Compute();
analogWrite(PWMPin, Output);

//Send data over serial for troubleshooting
if(millis()>serialTime){
  sendData();
  serialTime += 500;
}
}

void sendData(){
  Serial.print("Setpoint: ");
  Serial.print(Setpoint); Serial.print(" ");
  Serial.print("Input: "); Serial.print(Input);
  Serial.println();
}

```