Ball and Beam: Design and Control

Mary Torrico-Claure(*), Alexis Fuentealba-Orrego, Camilo Huanca-Mendoza, Ricardo-Franco Mendoza-Garcia

Escuela Universitaria de Ingenieria Mecanica
Campus Saucache, Universidad de Tarapaca
18 de Septiembre 2222, PO BOX 1010069
Arica, Chile

Abstract

A PID controller allows for precise control of the dynamics of a system during transitory and stationary states; a feature that can be useful to extend the motors life in a little robot or to equilibrate the liquid within containers in a huge cargo ship. This work describes the design of a *Ball and Beam* platform for testing control algorithms, and the implementation of a particular PID controller for it. The platform is made out of 2D-wooden parts, an Arduino UNO, a PING ultrasonic sensor, a standard servo motor and screws. The controller is implemented in Arduino language and includes proportional, integrative and derivative terms. The system is successfully stabilized; the ball can be positioned at any point within a range of distance. The platform is cheap, easy to replicate and allows for further testing of alternative control algorithms.

Keywords: Ball and Beam, PID control, PING sensor, Arduino

1. Introduction

This project is developed in the context of a control course of the Mechatronics career of the Universidad de Tarapaca. It consists on building and controlling a self-balancing system called Ball and Beam that must adjust the inclination of the beam in order to quickly position the ball at desired distances from the center (see Fig. 6).

Designs should be simple, cheap and strong. They should also utilize the equipment available at the University which include laser cutting machines, 3D-printers, and CNC, lathes and milling machines. For the control loop, electronics components permitted are Arduino boards and standard servomotors. Thus, PID controllers [1] must be programmed in Arduino language.

From a purely mechanical point of view, the first challenge is to design a transmission that converts the 180° rotation of the servomotor to a smaller rotation at the beam. Lever mechanisms were discarded due to possible asymmetries when tilting in one direction or another that could impact the performance of the controller. So, the first prototypes had a pulley transmission which did not work either due to sliding at the belts that introduced misalignment between servomotor and beam. The Ball and Beam is finally implemented using spur gears. The transmission has a good response to movement of the actuator, but helical gears [2] are recommended instead because movement is more uniform and lacks of backlash. Anyhow, the system is successfully stabilized; the ball can be positioned at any point within a range of distance.

2. Design

2.1. Structure

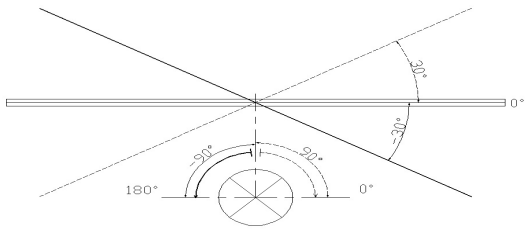
The structure must be simple and strong. A mechanical reduction must transmit a $\pm 90^o$ rotation at the servomotor to a $\pm 30^o$ inclination at the beam, as shown in Fig. 1a. A gear reduction 3:1 was then decided; number of teeth, step size, and gear module calculated [2]; and tooth profile drawn in AutoCAD[1] according to the 20° `geometric method`, as shown in Figs. 1b and 1c.

2.2. Control loop

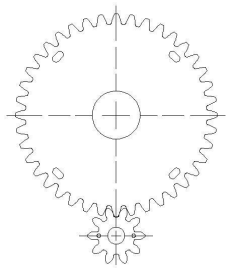
In order to close the control loop, an Arduino UNO board, a PING sensor and a Goteck GS-3630BB standard servomotor were used (see Fig. 2). The PING sensor would be placed at one end of the beam, as shown in

*Corresponding author. Tel.: +56 58 2205091; fax: +56 58 2205281. Email: `m.torricoclaure@gmail.com`

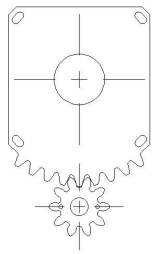
[1] http://www.autodesk.com



(a) Required transmission of angular movements.



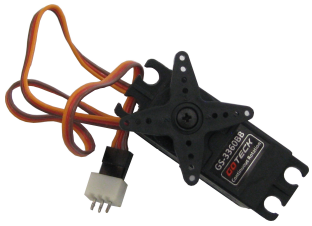
(b) Original design.



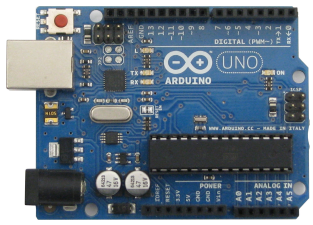
(c) Modified design.

Figure 1: (a) shows the requirement for the angular transmission, (b) shows the original idea for the gear design, and (c) shows a modification excluding unused teeth.

(a) PING ultrasonic sensor.



(b) Standard servomotor (image is only referential).



(c) Arduino UNO board.

Figure 2: Components to close the control loop.

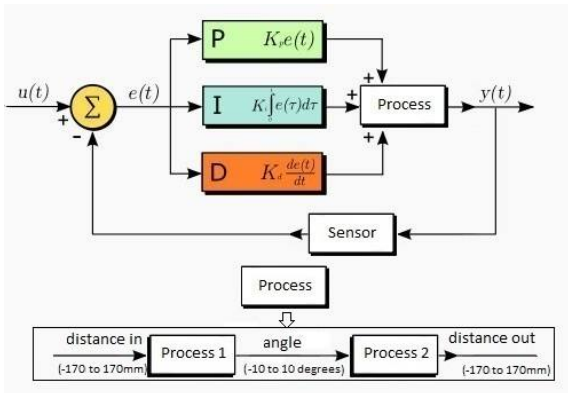


Figure 3: Block diagram with closed-loop control.

Fig. 4c, so the ultrasonic signal would travel through a narrow passage until it is reflected by the ball. The different components of the block diagram describing the dynamics of the system were also identified, as shown in Fig. 3. Finally, the PID algorithm was planned according to the discrete transfer function shown in Eq. 1, as provided by Bolton in [3].

$$X_n = K_P\, e_n + K_I\left(\frac{(e_n + e_{n-1})\, T_s}{2} + Int_{prev}\right) + K_D\left(\frac{e_n - e_{n-1}}{T_s}\right) \tag{1}$$

3. Results

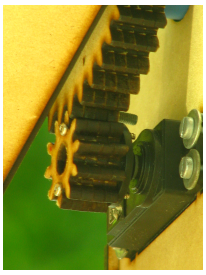
3.1. Structure

The 2D pieces were laser cut on 3mm thick plywood layers. All the cuts for the structure were made out of a single plywood layer (see Fig. 4a), but the gears were made out of four layers glued together in order to ensure proper engagement. The resulting gear thickness was then 12mm, as shown in Fig. 4b.

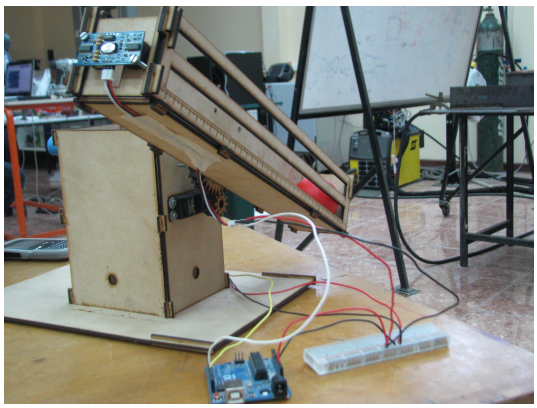
3.2. Control loop

As with any closed-loop system, the sensory feedback was crucial, but the original design presented undesired reflexions that forced some modifications. At first, exposed teeth besides the beam were covered by a shiny resin (see Fig. 4a) but then they were completed cut off. This modification improved feedback quality. The PID controller was implemented as the program shown in Fig. 5, and its gains were tunned according to the procedure indicated by Wescott in [4]: first K_D was set in the border of oscillation, then K_P was also set in the border of the oscillation, and finally K_I was set to a value that stabilized the whole system. Following this approach, the PID response was adequate but slightly slow. Set-point and debugging information was sent/received to/from the system through the serial USB interface provided by Arduino, as shown in Fig. 6.

(a) Single-layer assembly.



(b) Multi-layer gears.



(c) Side view assembly showing sensor and actuator.

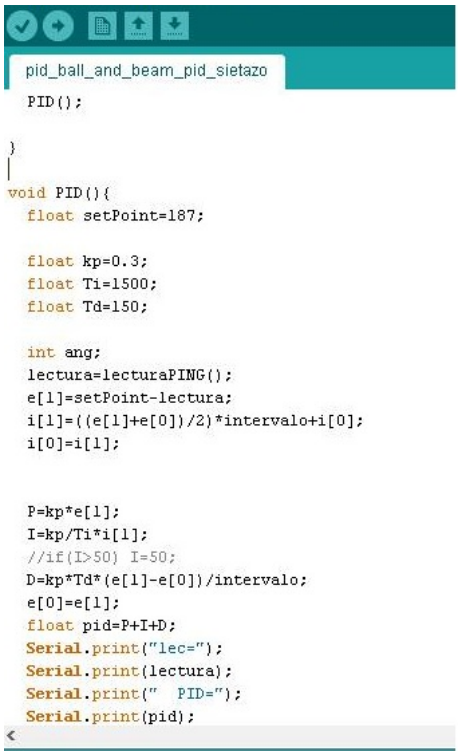
Figure 4: Assembly made of wood and screws.

4. Discussions

The shape of the ball has a huge influence in the quality of the sensor feedback. It was rather difficult to place the PING sensors at the right height to hit exactly the middle of the ball and thus to have a good estimation of distance to one extreme. Wide wheels were tested and the measurements quality improved considerably.

5. Conclusions and Future Work

A Ball and Beam system was built using cheap components and a laser cut. A PID controller for the system was programmed in Arduino which successfully controlled the ball distance according to set-point commands sent by an user from a console. The controller is precise but a bit slow. The most difficult part of the implementation was to obtain a good quality sensor signal. Due to that, a CMUcam4 sensors will be tried in a future project that, taking advantage of the 2D position feedback, will be a Ball and Plate system instead.



```
pid_ball_and_beam_pid_sietazo
  PID();

}

void PID(){
  float setPoint=187;

  float kp=0.3;
  float Ti=1500;
  float Td=150;

  int ang;
  lectura=lecturaPING();
  e[1]=setPoint-lectura;
  i[1]=((e[1]+e[0])/2)*intervalo+i[0];
  i[0]=i[1];


  P=kp*e[1];
  I=kp/Ti*i[1];
  //if(I>50) I=50;
  D=kp*Td*(e[1]-e[0])/intervalo;
  e[0]=e[1];
  float pid=P+I+D;
  Serial.print("lec=");
  Serial.print(lectura);
  Serial.print("  PID=");
  Serial.print(pid);
```

Figure 5: PID controller algorithm.

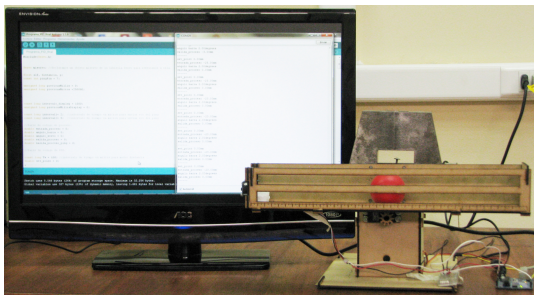


Figure 6: Complete ball and beam system.

References

[1] K. Ogata, Modern Control Engineering, 5th Edition, Prentice Hall, 2009.

[2] R. L. Mott, Machine Elements in Mechanical Design, 5th Edition, Prentice Hall, 2013.

[3] W. Bolton, Mechatronics: Electronic control systems in mechanical and electrical engineering, 4th Edition, Prentice Hall, 2009.

[4] T. Wescott, Pid without a phd, www.embedded.com (October 2000).