Ball Balancing PID System
Progress Review III - 4th April 2023

D/ENG/21/0021/MC | W L S Wickramaarachchi
D/ENG/21/0036/MC | K D W Silva
D/ENG/21/0045/MC | K W R M G S B Pilapitiya
F/ENG/21/6275/MC | Umubyeyi Pascaline

Since PRII on March 9th, we have designed and fabricated nearly the entirety of our system.

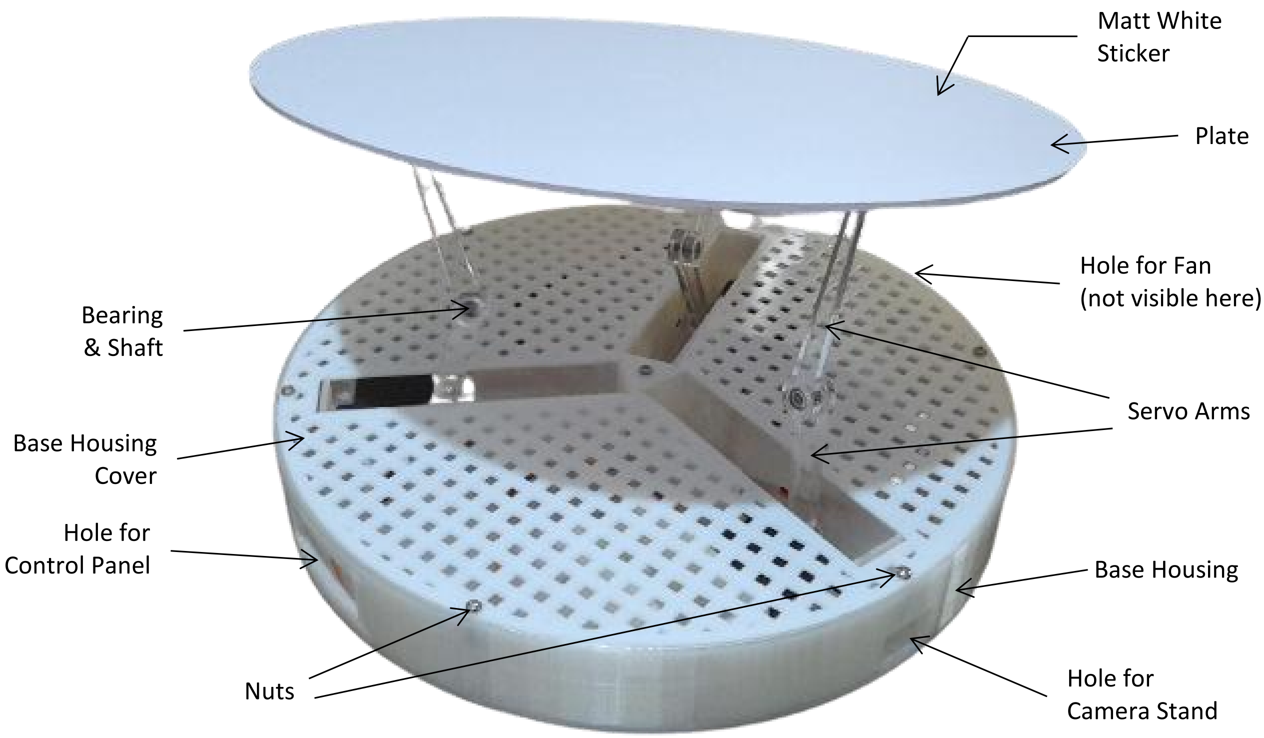

Figure 1: Mechanical Design of Ball Balancing PID System

Only the Camera Stand remains to be fabricated, and the program needs to be tested and optimised.

Mechanical Design

Solidworks was used to create a Base Housing, which was then 3D printed. The printer available at the FOE's Design Lab on the 2nd floor (a Makerbot METHOD) was not large enough to print the complete structure at once, so we had to divide it into seven sections, print them individually, and then glue them together. (This is illustrated on the following page.)

Figure 2: Solidworks Design of Base Housing

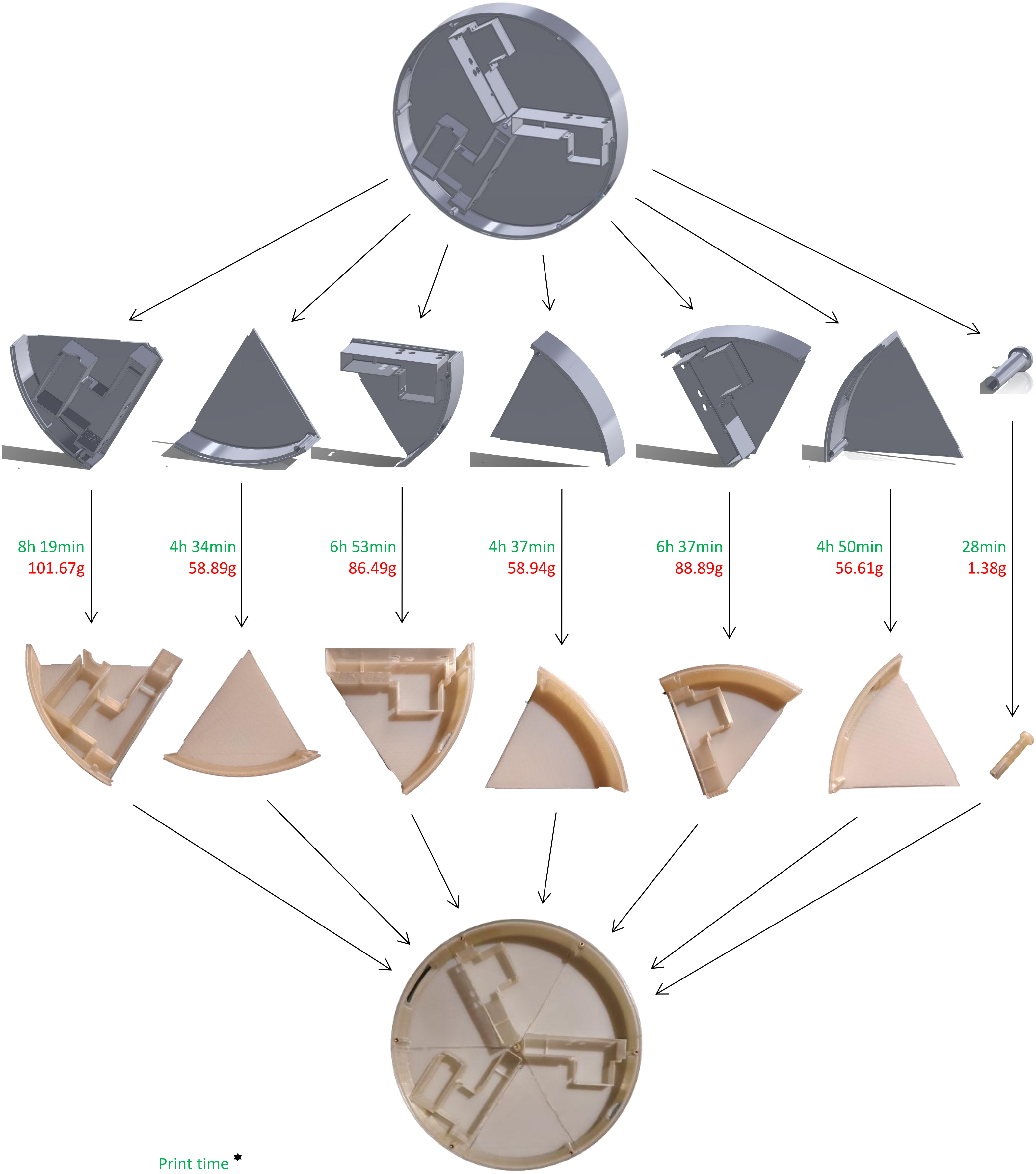

Figure 3: Printed & Assembled Base Housing

The material used for the print was Polylactic Acid (PLA), and the total mass of material used was **452.87g.** The total print time was **36h 18min**, over a period of **5 days**.

The Plate and Base Housing Cover were made from 2mm thick clear and white acrylic respectively.

A Matt White Sticker has been applied on to the plate. The plate couldn't be made of white acrylic since it would reflect too much light into the camera.

The Base Housing Cover has a series of square shaped holes on it. This, together with a Cooling Fan, will assist to ventilate and cool the components inside the Base Housing.

The Base Housing has 7 hexagonal holes for inserting and super-glueing 1cm long metal spacers. Nuts that go through holes in the Base Housing Cover and latch into the threads in the spacers secure the Base Housing Cover to the Base Housing.

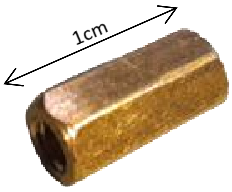

Figure 4: Spacer

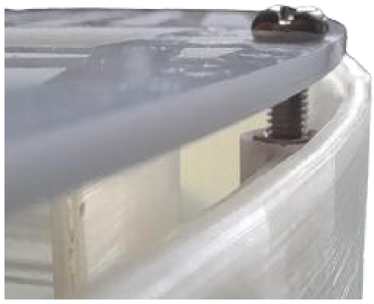

Figure 5: Nut securing the Base Housing Cover on to the Base Housing

The Base Housing also contains holes for attaching a Control Panel (which will contain a switch, the power inlet, and an indicator LED), a Cooling Fan, and the Camera Stand. (These are visible in Figure 1.)

Inside the Base Housing, there are provisions for fastening the Servo Motors to the structure.

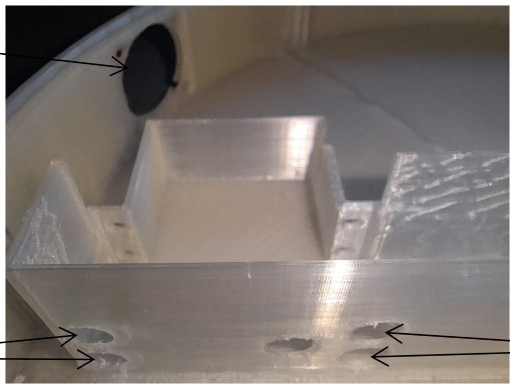

Figure 6: Provision for attaching a Servo Motor

The Servo Arms were fabricated by laser cutting clear acrylic. They attach onto the Servo Motors' gears via a gear pattern cut into them.

Figure 7: Servo Arm

The Servo Arms attach on to the Plate through a Ball & Socket Joint between the N20 Universal Wheel Ball, and 3D printed sockets.

Figure 8: 3D Printed Socket

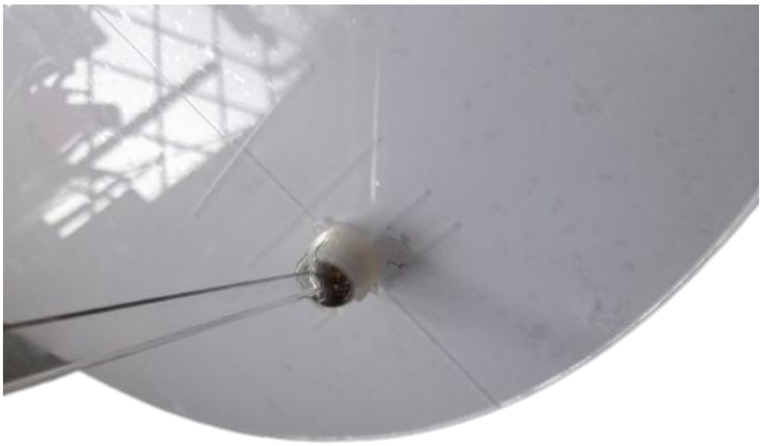

Figure 9: Ball & Socket Joint on Plate

Programming

While fabrication was still underway, the code was written to work for the manual inputing of x & y co-ordinates.

The program works as follows:

Since both the Proportional (P) and Derivative (D) variables of the PID control the same output variable, we manually define a proportionality factor variable "k" to control the P & D variables. Once we start testing with the fabricated mechanical design, we will perform a trial & error test to determine the optimal value for k.

```
t = 1              # delay
rt = math.sqrt(3) # root of 3, used for determining which slice the ball is on
st = 1             # time spent stationary at a moment
r = 10             # mock radius of ball in pixels
k = 1              # proportinality factor for P and D
x = 0
y = 0
xp = 99            # previous x values (initialized to avoid a value error)
yp = 99            # previous y values (initialized to avoid a value error)
v = 1              #initialized value
```

For manual testing, we used a while loop to continuously input x & y coordinates.

```
while True:
    x = float(input("Enter X co-ordinates : "))
    y = float(input("Enter Y co-ordinates : "))

    d1 = 0
    d2 = 0
    d3 = 0
```

The Plate is divided into six equal slices, and using the x & y coordinates and a series of if conditions it is determined in which slice the Ball is present in. This decides which motors are to respond (only two out of three of the motors respond at any given time), and by how much (d1,d2,d3).

```
    if y > rt*x and y > -rt*x:
        d1 = y/rt + x
        d3 = -y/rt + x
    if y < rt*x and y>0:
        d1 = (y*rt)/2
        d2 = -x + y/rt
    if y >-rt*x and y<0:
        d3 = (y*rt)/2
        d2 = -x - y/rt
    if y < rt*x and y < -rt*x :
        d1 = - y/rt - x
        d3 = y/rt - x
    if y > rt*x and  y<0 :
        d1 = -(y*rt)/2
        d2 = x - y/rt
    if y < -rt*x and  y>0 :
        d3 = -(y*rt)/2
        d2 = x - y/rt
```

PID Variables:

1. Proportional Variable
   Use the x & y co-ordinates to calculate values for d1, d2, and d3, which represent the origin vector for the ball to the center, on a plane with axis' parallel to the Servo Arms. Furthermore, the values t1, t2, and t3 (angle values given to the respective motors) are directly proportional to d1, d2, and d3.

2. Integral Variable
   x, y, xp, and yp values are used to calculate the number of frames the ball remains stationary (the variable st). This is done to make the orientation of the Plate change to force the ball to move towards the center, in the event that the ball stops at anywhere but the center.

3. Derivative Variable
   x, y, xp, and yp values are used to calculate the speed of the ball (v). The t1,t2 and t3 values are inversely proportional to the v. This helps slow the ball down to avoid slipage, and gives the system time to process and avoid the ball going past the center.

```
  if xp != 99 and yp != 99 :
        v = math.sqrt((xp-x)**2+(yp-y)**2)/t  # speed
        if xp == x and yp == y:
            st = st + 1  # accumaltion of time spent stationary
        else:
            st = 1       # resetting st when ball moves
    yp = y
    xp = x
```

The final equation to calculate t1, t2, and t3 values using the above mentioned processes, and normalize the output to a value between 0 and 90 degrees (which is going to be the operating range of the Servo Motors), is as follows:

```
    t1 = ((r+(d1*k*(st/100))/v)/(r))*45
    t2 = ((r+(d2*k*(st/100))/v)/(r))*45
    t3 = ((r+(d3*k*(st/100))/v)/(r))*45
```

If the values of t1, t2 or t3 are greater than 90 or less than 0, this suggests that the ball is outside the plate, and t1,t2, and t3 will be reset to 45 degrees (default angle).

```
    if t1 > 90 or t2 > 90 or t3> 90 or t1<0 or t2<0 or t3< 0 :
        t1=45
        t2=45
        t3=45
```

- END -