DARUMA PLATE

(BALL BALANCING ON A PLATE)

VEDNARAYAN IYER , LAUKIK HASE , AMEYA APTE , PARTH SHAH

MECHANICAL DEPARTMENT

SRA VJTI MUMBAI-400019

Email ID :

Vednarayan Iyer- [vednarayan29@gmail.com](mailto:vednarayan29@gmail.com)

Laukik Hase - [laukik17@gmail.com](mailto:laukik17@gmail.com)

Ameya Apte – [apte.ameya8@gmail.com](mailto:apte.ameya8@gmail.com)

Parth Shah - [bholushah99@gmail.com](mailto:bholushah99@gmail.com)

***GENERAL OVERVIEW & SIGNIFICANCE:***

The project is all about balancing the ball on a plate using PID control system using arduino, Image processing using OpenCv python. This project is actually to understand the principle of any balancing system(drone, robotic car, basic human robot, etc.) .

***OUR PROJECT EXPLANATION IN SIMPLE WAY:***

Our project basically consists of a camera , Arduino board , CP , programmer and two servo motors

***FLOW OF OUR PROJECT:***

**NOTE: All the codes related to image processing, image thresholding and serial communication are written in python and all the codes related to PID control are written in Arduino.**

First the camera captures the frame of the plate and we take the pixel values of the center of the plate by writing a small basic code in python .This pixel values of the center of the plate is set as the setpoint.

Camera actually takes the video dynamically during runtime and gives input to the laptop as frames (We have made use of a webcam which gives 17 frames per second).

According to the image processing program only one contour is made around the ball which is placed on the plate, according to this contour center of the ball is located and each time the ball acquires a new position the center of the ball changes i.e. the center of the ball keeps on changing dynamically during runtime .

Hence error is calculated by subtracting the center of the ball from the setpoint. This error calculated is continuously sent via CP to the Arduino board. Hence each time the ball moves the center of the ball keeps on changing hence the error keeps on changing. Hence all these errors are continuously sent one after the other via CP to the Arduino board .

The errors are sent as soon as the ImgProcess.py gets executed .Before executing this program the PID.ino program is uploaded to the Arduino board using a programmer .This is because the work to be done on the errors i.e. converting errors into servo angles using the formula **const\*error=servo angle** is done by the PID.ino program, hence PID.ino program is to be executed and uploaded first before executing the ImgProcess.py program.(means basically what happens is the PID.ino program gets stored in the microprocessor present on the Arduino board and when the errors are received by the microprocessor it starts executing the steps written in the PID.ino program on the errors.

And hence the output from the microprocessor i.e. servo angle is given to the servo motor as input according to which the servo motors change there angle and hence the plate gets tilted. The correct value for ‘const’ in the formula **const\*error=servo angle** is to be found out by trial and error method for which the ball on the plate gets balanced.







***COMPONENTS, MATERIALS AND SOFTWARES USED:***

1. Two servo motors.
2. Aluminium L-shape rods.
3. Wooden plank for base and the plate on which the ball is going to get balanced.
4. Buck convertor (Used here as a voltage regulator)
5. CP(for serial communication)
6. Programmer(to upload program from laptop to arduino board)
7. Laptop with minimum three USB ports
8. Arduino board
9. A ping pong ball which is going to get balanced.

(NOTE: Reason behind using light weighted ball is that the inertia offered by it is less hence its response time is less i.e. it gives fast response.)

1. Female to female jumpers.
2. Perf board(on which components can be mounted by means of soldering)
3. Male bergstrips
4. Screw terminals
5. Web camera(used for image processing)
6. Ample amount of screws and lock nuts
7. Python and opencv
8. Arduino compiler

***MECHANICAL DESIGN:***

The design actually comprises of a wooden plank used as the base on which two servo motors are mounted using L-shape aluminium rods of desired size.

There is a central pivot rod on which the plate (which is going to balance the ball) is mounted . An aluminium plate is mounted on the servo horn which in turn gets connected to each servo motor respectively.

Two pieces each with a slot of desired dimensions are connected to the plate(on which ball is going to get balanced ) which in turn gets connected to the plate mounted on the servo motor with the help of screw (this actually makes up the sliding mechanism due to which the plate gets a particular tilt for different servo angles ). All the above structures are shown below,



***CODING PART:***

This includes image processing ,PID control and serial communication.

***1) PID CONTROL:***

What PID actually means is ‘learning from mistake’. It is basically a inherent property that every human has . Imagine if you are practicing to juggle three balls. Now the first time when you try juggling, off course you won’t get it correct .

Then you keep trying again and again and each time you do it , you correct yourself. Now in case of juggling P(proportional) is the speed with which you throw your ball. While juggling you realize that there is always some constant-error which you aren’t able to avoid.

However by noticing the time the ball takes to go up and down and noting the exact time for perfect juggling you can avoid that steady error, this is the I(integral) term.

By this time you would have understood how to juggle but sometimes you drop the ball. You can avoid this by comparing the instances of success and failures. This is the D (derivative) term.

We deﬁne the SP (SetPoint) at coordinate (160,160) which is the center of the plate. The disturbance is introduced by exerting a force on the ball alongside of an arbitrary direction. The sensor is the camera on top of the plate, and PV (Process Variable) refers to actual ball coordinate that is calculated through image processing. To implement the above controller in arduino programming language the diagram needs to be converted into its mathematical formula and then an algorithm needs to be written

Fig. 7. General PID control block diagram that can be fully implemented into machine code.

Equation (1) shows the general PID control formula.

**u(t) = Kp.e(t) + Ki + Kd e(t)**

Equation (1) consists of the summation of three terms. Each term has a coefﬁcient with the following deﬁnition:

Kp: Proportional coefﬁcient that is proportional to the error.

Ki: Integral coefﬁcient that accounts for past errors.

Kd: Derivative coefﬁcient that predicts future errors.

The proportional term is just a simple multiplication that can be easily implemented into machine code using the arduino programming language. The integration term can be calculated using Euler integration technique by deﬁning a small ﬁxed time interval which is named dt. Image processing rate is at 17 FPS (Frame Per Second), it means that it approximately produces results which arrive every 58.82ms.

The variable dt is set to 0.015 to ensure that upon arrival of each image processing result (as a ball coordinate) the system advances correctly. Therefore each run of the loop in PID control is considered to take 58.82ms. In our case we have only used the P and D term. Our system consists of two servo motor, thus we have to apply PID two times on both the servo respectively. Now each term has its own constant. In this case Kpx, Kpy, Kdx and Kdy. These constants are found out by tuning your system. By tuning I mean you can take a calculated value of Kpx first calculation is like this Kpx\*max. Error = max angle.

The value of max. Error and max. Angle is found out based on your system. In our case , using IP we found out the max possible error and max angle we decided was around 20-25 degrees. While doing this keep your Y-servo off and seeing the behaviour of ball using only 1 servo. Now for Kdx term , you have to consider how much your ball is overshooting .

The D term is subtracted from the P term to decrease overshooting . Hence the Kdx term is negative. By considering these constants tune your system by changing these constants until your ball balances on the central line. Now turn on your second servo in addition to the first one. Repeat the same tuning process on Y-constants until the ball is balanced .

***2) IMAGE PROCESSING:***

So, we are using webcam to track the position of ball on the plate.

Now, there are two methods two track the motion of ball. In both the method OpenCV on python is used.

* **USING A SINGLE COLORED BALL ON A BLACK COLORED PLATE:**

In this method we use the function cv2.inRange() to detect a particular color on frame. Then using findContours() and drawContours() functions made a outline on the ball.

Then by using cv2.minEnclosingCircle(cnt) function found out the coordinates of ball. So, our 1st idea was to use a green ball on black background. However it was hard to find a ideal green seamless ball. So we dropped that idea

* **USING ANY COLORED BALL EXCEPT BLACK ON BLACK BACKGROUND:**

In this method we first convert the frame to grayscale and then by using threshold method convert all the colors except black to white. So by this method we can track any ball of any color. Now similar to the previous method found the coordinates of the ball.

***3) SERIAL COMMUNICATION:***

We are applying PID using arduino however our IP is on Python and for PID calculation we need the coordinates of the ball, so we have to create a communication between arduino and Python. For doing so we used serial communication.For this we have import pyserial on python and create an object to invoke the communication between them.You need to connect CP to your PC . Connect CP’s ground ,RX and TX to you arduino board using Jumpers.

***ELECTRONICS PART:***

We have made use of arduino board for processing purpose. We have also used a buck converter to convert 12 volts supply from the battery into 5 volts output(as servo motor requires 5 volts as supply).Here one can make use of IC 7805 which acts as voltage regulator . This IC’s must be such that if here are two servo motors then four 7805 IC’s are required in parallel .

***PROBLEMS FACED :***

1. Accurate mounting of servo motors and web camera.
2. Shorting of components.
3. Breakage of touchscreen.

***SOLUTIONS:***

1. The servo motors that are mounted must be perfectly placed and the camera center must be exactly aligned with the center of the plate.
2. Make sure that the soldered part of the component doesn’t come in contact with aluminium or any other metallic part so that shorting of the component can be avoided.
3. Touchscreen is very much fragile so touchscreen must be handled with extreme care and if possible try to avoid use of touchscreen.
4. Make sure that the laptop which you are using doesn’t get wet as if it is used in this condition then its motherboard may get short and the laptop may get heavily damaged.
5. Make sure that every component is perfectly insulated so that shortening can be avoided.
6. While using buck convertor or any other voltage regulator make sure that you don’t cross connect the wires with terminals of the battery.

***FOR CODES AND VIDEOS CLICK BELOW:***

Video : <https://m.youtube.com/watch?t=4s&v=n7utjRaWdz8>

***REFERENCES:***

1. ieeexplore.ieee.org/document/7419072



 

