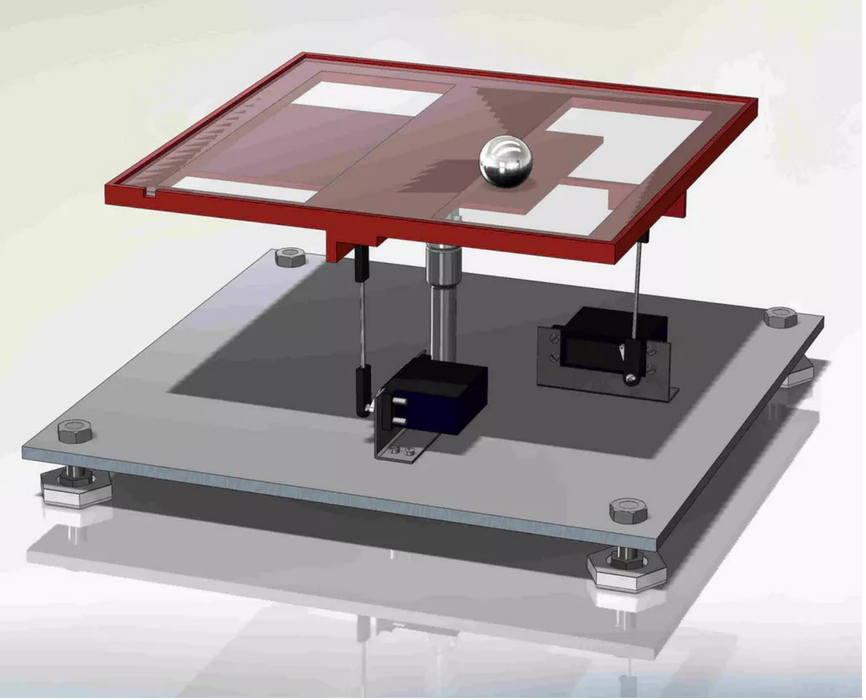
Ball on Plate Balance

Maya Nasr

MIT-IIT Robotics Program 2017

Ball on Plate Lab

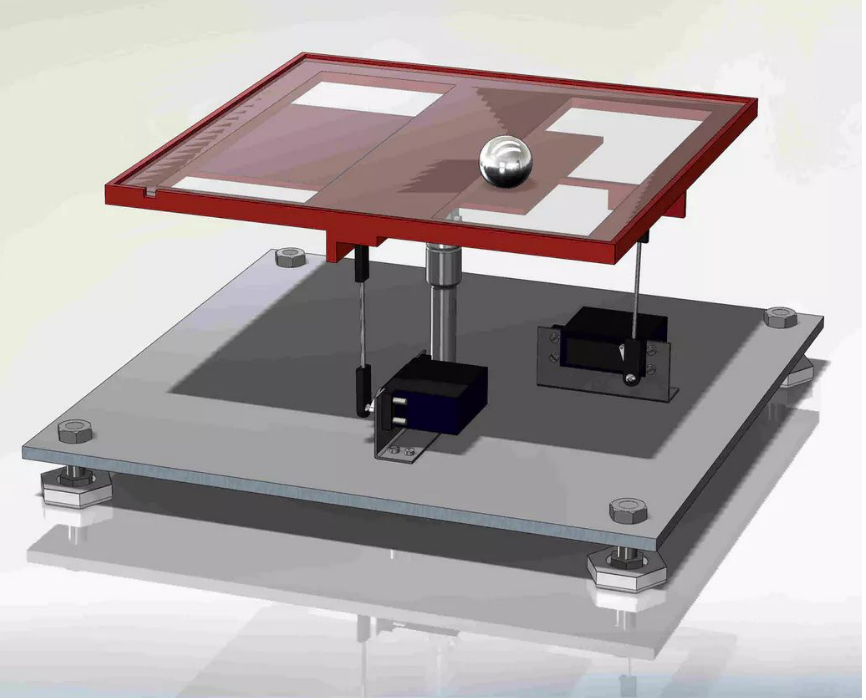
- The Ball on Plate system is the 2-D (two dimensional) extension of the Ball on Beam system.



Ball on Plate Lab

- The Ball on Plate system is the 2-D (two dimensional) extension of the Ball on Beam system.

- It has an x and y-dimension which allows the ball to roll along different paths.



Libraries

- Write an Arduino code to balance the ball on a plate.
- You can use the following Arduino libraries:
  - Servo library `<Servo.h>`

Libraries

- Write an Arduino code to balance the ball on a beam.
- You can use the following Arduino libraries:
  - Servo library `<Servo.h>`
  - Touchscreen library "TouchScreen.h"
    - TouchScreen ts = TouchScreen(XP, YP, XM, YM, 711);

Libraries

- Write an Arduino code to balance the ball on a beam.
- You can use the following Arduino libraries:
    - Servo library `<Servo.h>`
    - Touchscreen library "TouchScreen.h"
        - TouchScreen ts = TouchScreen(XP, YP, XM, YM, 711);
        - TSPoint p = ts.getPoint();

Libraries

- Write an Arduino code to balance the ball on a beam.
- You can use the following Arduino libraries:
  - Servo library `<Servo.h>`
  - Touchscreen library "TouchScreen.h"
    - TouchScreen ts = TouchScreen(XP, YP, XM, YM, 711);
    - TSPoint p = ts.getPoint();
    - x -coordinate is(p.x); y -coordinate is (p.y);

Libraries

- Write an Arduino code to balance the ball on a beam.
- You can use the following Arduino libraries:
    - Servo library `<Servo.h>`
    - Touchscreen library "TouchScreen.h"
        - TouchScreen ts = TouchScreen(XP, YP, XM, YM, 711);
        - TSPoint p = ts.getPoint();
        - x -coordinate is(p.x); y -coordinate is (p.y);
        - Pressure is (p.z) (Note: ts.pressureThreshhold)

Libraries

- Write an Arduino code to balance the ball on a beam.
- You can use the following Arduino libraries:
  - Servo library `<Servo.h>`
  - Touchscreen library "TouchScreen.h"
    - TouchScreen ts = TouchScreen(XP, YP, XM, YM, 711);
    - TSPoint p = ts.getPoint();
    - x -coordinate is(p.x); y -coordinate is (p.y);
    - Pressure is (p.z) (Note: ts.pressureThreshhold)
  - PID library `<PID_v1.h>`

PID library `<PID_v1.h>`

PID()

Creates a PID controller linked to the specified Input, Output, and Setpoint.

PID library `<PID_v1.h>`

PID()

Creates a PID controller linked to the specified Input, Output, and Setpoint.

Syntax

PID(&Input, &Output, &Setpoint, Kp, Ki, Kd, Direction)

For reference, in a car, the Input, Setpoint, and Output would be the speed, desired speed, and gas pedal angle respectively.

PID library `<PID_v1.h>`

PID()

Creates a PID controller linked to the specified Input, Output, and Setpoint.

Syntax

PID(&Input, &Output, &Setpoint, Kp, Ki, Kd, Direction)

For reference, in a car, the Input, Setpoint, and Output would be the speed, desired speed, and gas pedal angle respectively.

Parameters

Input: The variable we're trying to control (double)
Output: The variable that will be adjusted by the pid (double)
Setpoint: The value we want to Input to maintain (double)
Direction: Direction the output will move when faced with a given error. DIRECT is most common.

Compute()

Description

Contains the pid algorithm. it should be called once every loop(). Most of the time it will just return without doing anything. At a frequency specified by SetSampleTime it will calculate a new Output.

Compute()

Description

Contains the pid algorithm. it should be called once every loop(). Most of the time it will just return without doing anything. At a frequency specified by SetSampleTime it will calculate a new Output.

Syntax

Compute()

Compute()

Description

Contains the pid algorithm. it should be called once every loop(). Most of the time it will just return without doing anything. At a frequency specified by SetSampleTime it will calculate a new Output.

Syntax

Compute()

Parameters

None

Returns

True: when the output is computed

False: when nothing has been done

SetMode()

Description

Specifies whether the PID should be on (Automatic) or off (Manual.) The PID defaults to the off position when created.

SetMode()

Description

Specifies whether the PID should be on (Automatic) or off (Manual.) The PID defaults to the off position when created.

Syntax

```
SetMode(mode)
```

SetMode()

Description

Specifies whether the PID should be on (Automatic) or off (Manual.) The PID defaults to the off position when created.

Syntax

SetMode(mode)

Parameters

mode: AUTOMATIC or MANUAL

SetOutputLimits()

Description

The PID controller is designed to vary its output within a given range.

Syntax

SetOutputLimits(min, max)

SetSampleTime()

Decription

Determines how often the PID algorithm evaluates. The default is 200mS.

SetSampleTime()

Decription

Determines how often the PID algorithm evaluates. The default is 200mS.

Syntax

SetSampleTime(SampleTime)

SetSampleTime()

Decription

Determines how often the PID algorithm evaluates. The default is 200mS.

Syntax

SetSampleTime(SampleTime)

Parameters

SampleTime: How often, in milliseconds, the PID will be evaluated. (int>0)

Balancing Ball on Beam Plate Code Steps

- Include the libraries you're using
- Define servo pin, Kp, Ki, Kd, Setpoint, Input, Output, and ServoOutput
- Initialize your PID object and your Servo
- Start the `setup()`
- After you finish your `setup()`, move to the `loop()`