Indian Institute of Technology, Kharagpur

"BALL BALANCE" Using PID control

A project by Group 6

ANIKETH SRIDHAR GUND	21ME10013
SAKSHAT ARYAN	21MI31021
ANKIT KUMAR	21PH10003
MUDAVATH MOUNIKA	21EX20001

Section -12

June, 2022

Abstract

The ball balancing system in a 1-degree of freedom platform is of unique importance in understanding the control system applications. It is a platform to test and identify different aspects of controls, as the non-linearities increases with the degree of freedoms. In this project we assumed only 1 degree of freedom. So far many techniques have been applied to sense the position of the ball on a plate in real time, most common of them is touchpad and audio video camera system. This report describes the design, development and control strategy of balancing a ball on a assembly using low cost ultrasonic sensor. These Ultrasonic sensors are triggered by Ultrasonic sound waves, which are reflected back to the sensor. This way the sensor calculates the distance. To balance the ball, a motor is used along x- axis. Hence the system utilizes one set of independent control mechanism, operating in isolation for one axis. *Arduino Uno R3* ***is used to run the assembly according our implemented code. it makes this controller ideal for the purpose. Dynamic modeling of the system yields the digital controller capable of balancing the ball in any of the desired points, on the assembly. Although the system becomes quite discrete but still it provides sufficient basis for implementing different control strategies and investigating different system parameters such as actuation mechanism, sensors, controller design and experimental testing, under the predefined condition of the ball diameter. The objective behind the Ball balance mechanism is, it can be implemented in robots, to balance them on two wheels or 1 wheel.***

Contents

Chapter 1

1.1 Introduction

This project "PID CONTROLLED BALL BALANCE", mainly focuses on balancing a ball on one particular axis (in this project it is *x-axis*). Control theory is a sub-field of mathematics operating within the area of dynamic systems. The main purpose of developing a control model is to either stabilize an unstable system or modulating the desired response and outcome of a given system. Furthermore, in this project, control theory will be applied within the area of engineering, more specifically mechatronics. Mechatronics is defined as multidisciplinary field of engineering that includes a combination of systems engineering: Mechanical, Electrical, control, computer engineering.

Automatic control theory

Although the disciplines within the area of automatic control theory varies, the theory and methods of analysis do remain, to some extent, unaltered. However, with various tasks of application the choice of automatic control design approach may vary. The controller used in the project is a proportional-integral-derivative controller, also referred to as a PID controller. The proportional part compensates for disturbances, but does not eliminate their eect. The integral part on the other hand does reduce the eect of the disturbances, but may cause issues with regards to system stability. Moreover, the derivative part of the controller does improve stability margins but at the expense of increased measurement errors.

1.2 Objectives

- Balancing a ball using sensors and programmable circuit boards
- This Balancing system can be assembled in robots
- To achieve perfect balance using an ultrasonic sensor which uses distance as a main parameter for working.
- Make this project an affordable

Some Important Remarks

- In this project, we will be balancing the ball in one particular axis only i.e in tis case the ball balances along x-axis only.
- We used one servo motor to control the angular displacement of the assembly
- The microcontroller converting data input into velocity and acceleration of the ball, sending control signals to a pair of servo motors. The servo motors, responding to signal from the microcontroller, adjusting in accordance with the control signal generating an angular displacement.

1.3 Why ball balancing project?

The ball balancing plays a very important role in mechatronics and robotics. Also, control theory is a very important concept in many parts of technology. Many models and developments have begun to emerge based on this theory. Just as each model is based on a theory, the ball balancing project is based on control theory. Balancing the ball placed on a table in the desired position is one of the most important examples of control theory.

1.4 Control theory :

In simple words control theory is the theory that states behaviour is caused not by outside stimuli, but by what a person wants most at any given time.
Objective: Control theory is concerned with the analysis and design of a closed-loop control system.
Analysis: Closed-loop system is given to determine characteristic or behaviour.
Design: Desired system characteristics or behaviour are specified → configure a closed-loop system.
Definition: A closed-loop system is a system in which certain forces (inputs) are determined, at least in part, by certain responses of the system(outputs).

Definitions:

– The system for measurement of a variable (or signal) is called a sensor (the ultrasonic sensor)
– A plant of a control system is the part of the system to be controlled. (the ball)
– The compensator (or controller) provides satisfactory characteristics for the total system. (servo motor)

Two types of control system:

– A *regulator* maintains a physical variable at some constant value in the presence of perturbances.
– A *servo mechanism* describes a control system in which a physical variable is required to follow or track some desired time function (originally applied in order to control a mechanical position or motion).

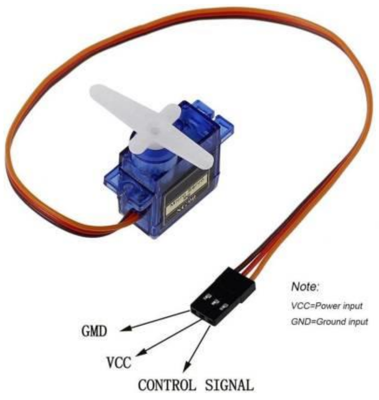
2.0 Our Model:

List of components used in making this model are given below:

Arduino Uno R3:

The Arduino Uno R3 is a microcontroller board based on a removable, dual-inline-package (DIP) ATmega328 AVR microcontroller. It has 20 digital input/output pins. Programs can be loaded on to it from the easy-to-use Arduino computer program

Servo Motor:



A servomotor (or servo motor) is **a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration**. It consists of a suitable motor coupled to a sensor for position feedback. Servomotors are used in applications such as robotics, CNC machinery, or automated manufacturing

Ultrasonic distance sensor

An ultrasonic sensor is **an instrument that measures the distance to an object using ultrasonic sound waves**. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity

Jumper wires (m-m, m-f, f-f)

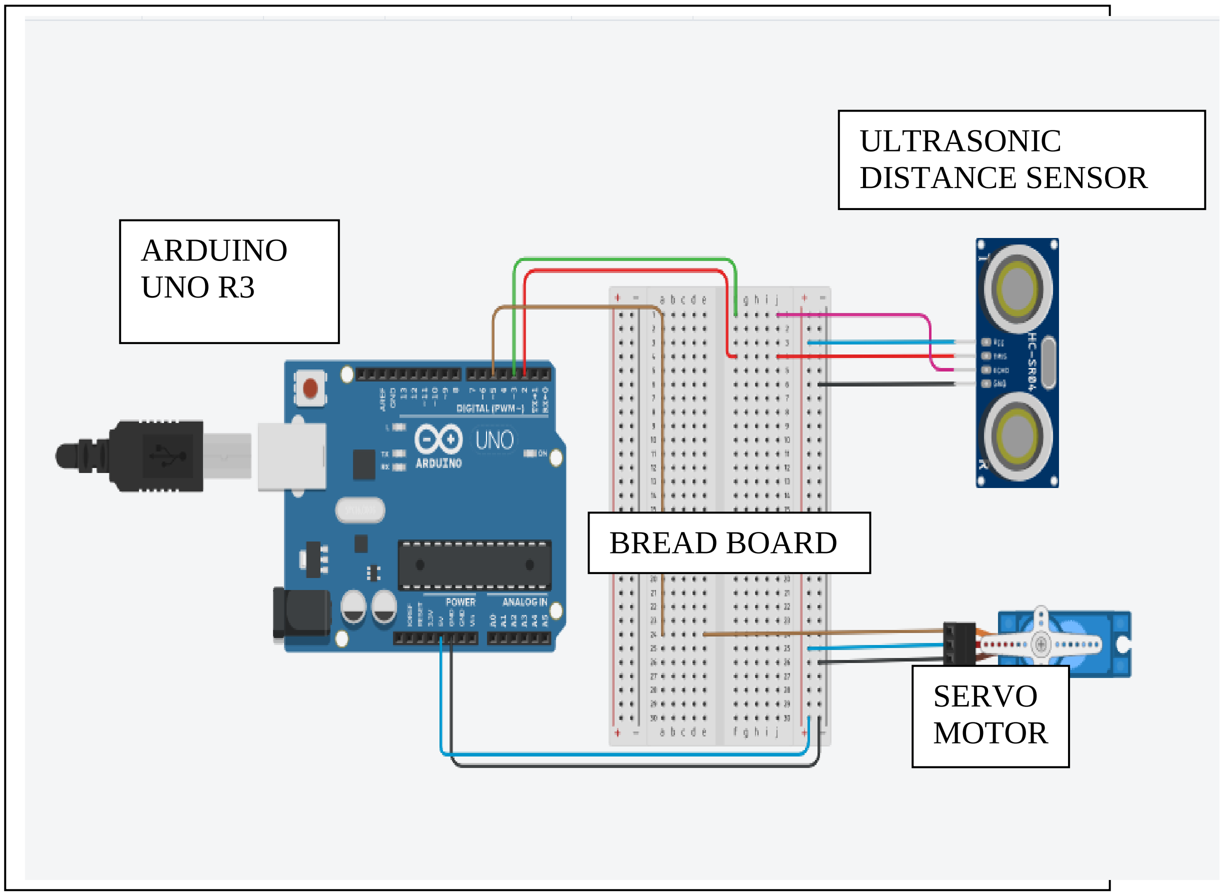
A **jump wire** (also known as **jumper**, **jumper wire**, **DuPont wire**) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

2.1 Circuit modelling:

For circuit modelling, we have used TINKER CAD software.
Tinker is an easy to use open source software. It is mostly used for simulation of electronic circuit boards, to check if they work according our way.

By using tinker cad software, we were able to simulate our circuit and also checked if the code written runs accordingly or not.

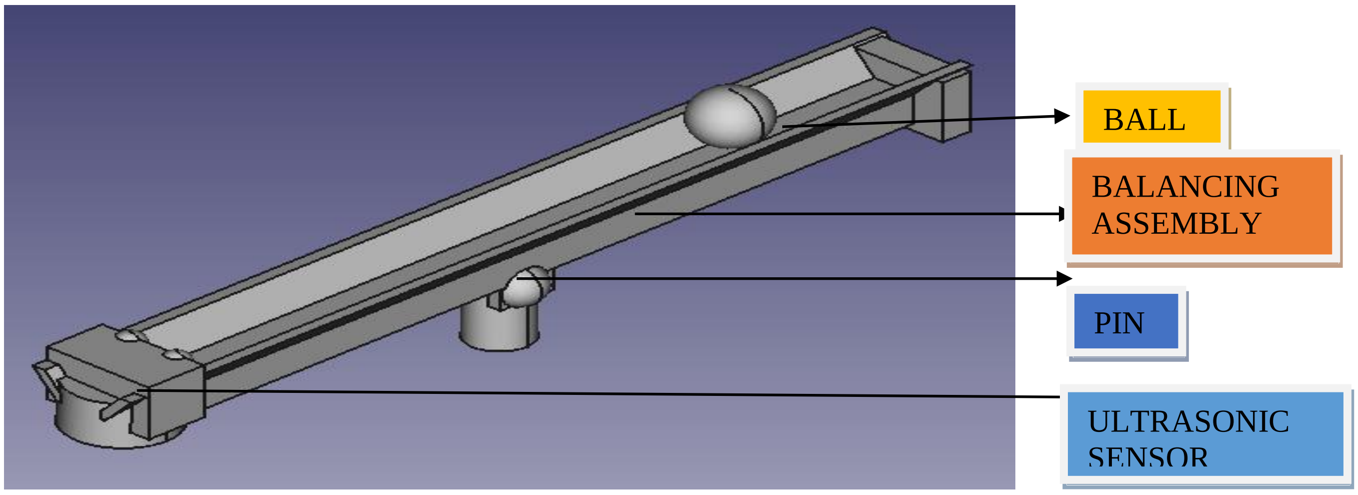
Tinker cad model is shown below:



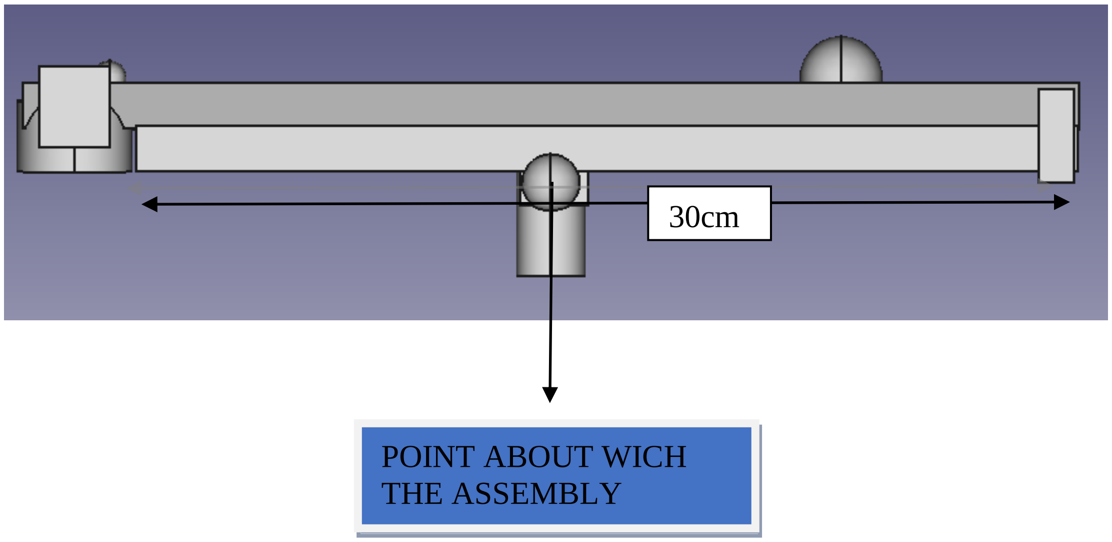
2.2 Free Cad modelling:

FreeCAD is a general-purpose parametric 3D computer-aided design (CAD) modeler and a building information modeling (BIM) software application with finite element method (FEM) support. It is intended for mechanical engineering product design but also expands to a wider range of uses around engineering, such as architecture or electrical engineering. FreeCAD is free and open-source, under the LGPL-2.0-or-later license, and available for Linux, macOS, and Windows operating systems. Users can extend the functionality of the software using the Python programming language.

Free cad model of the Ball Balance:

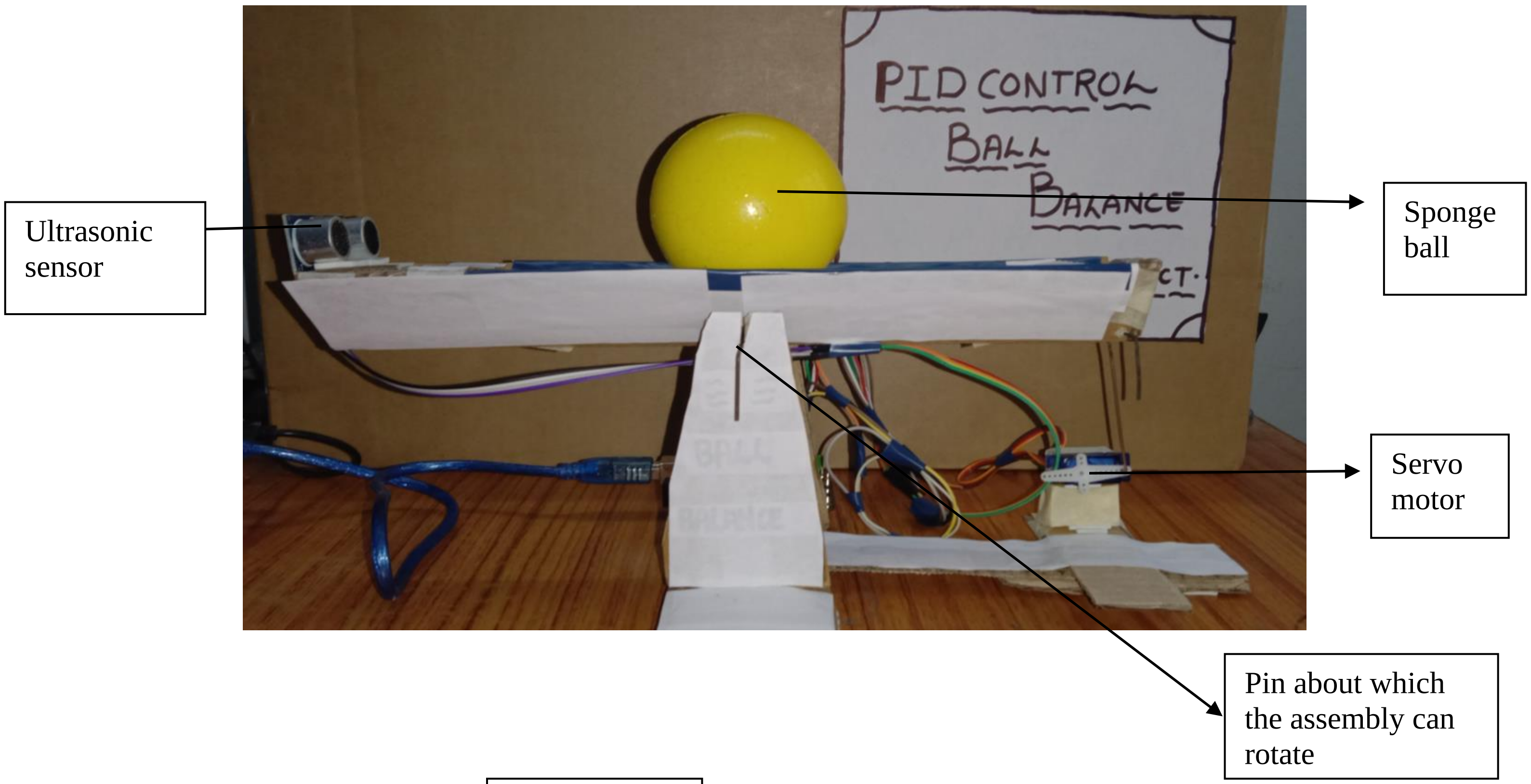


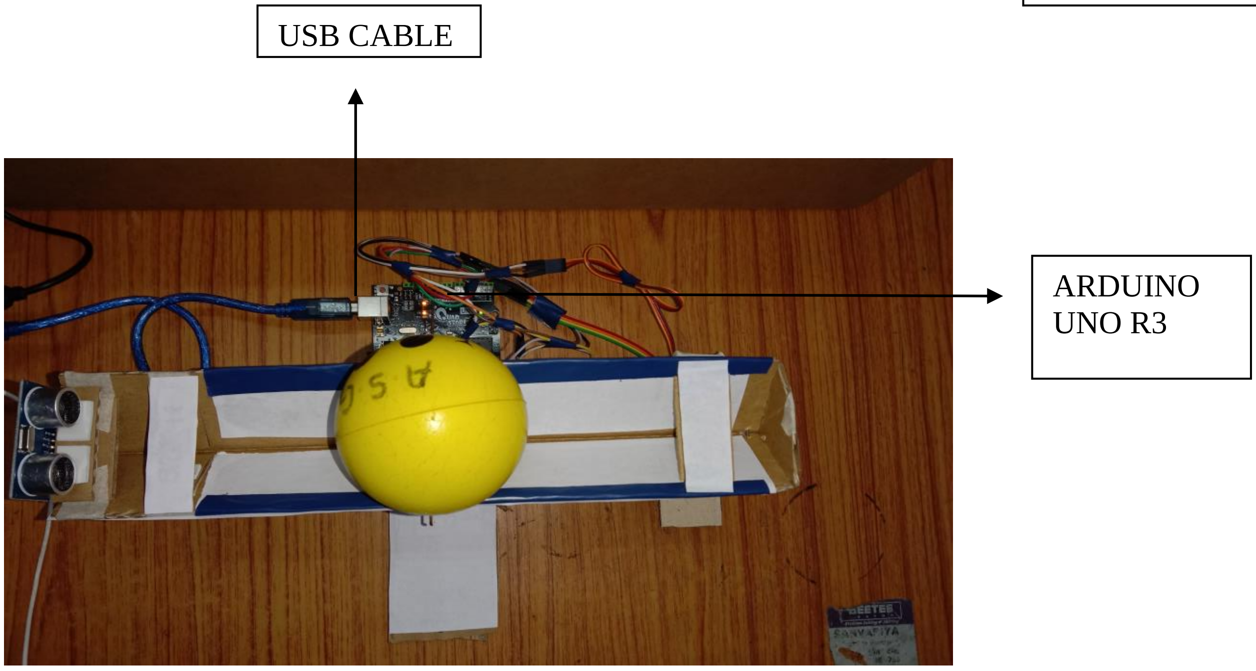
Front view:



2.3 Hardware Model of the project:

We used cardboard for the assembly, support and a hard steel wire for controlling the movement of the assembly with the help of servo motor.





3.0 About PID control Mechanism:

The basic principle of the PID control scheme is to act on the variable to be manipulated through a proper combination of three control actions:

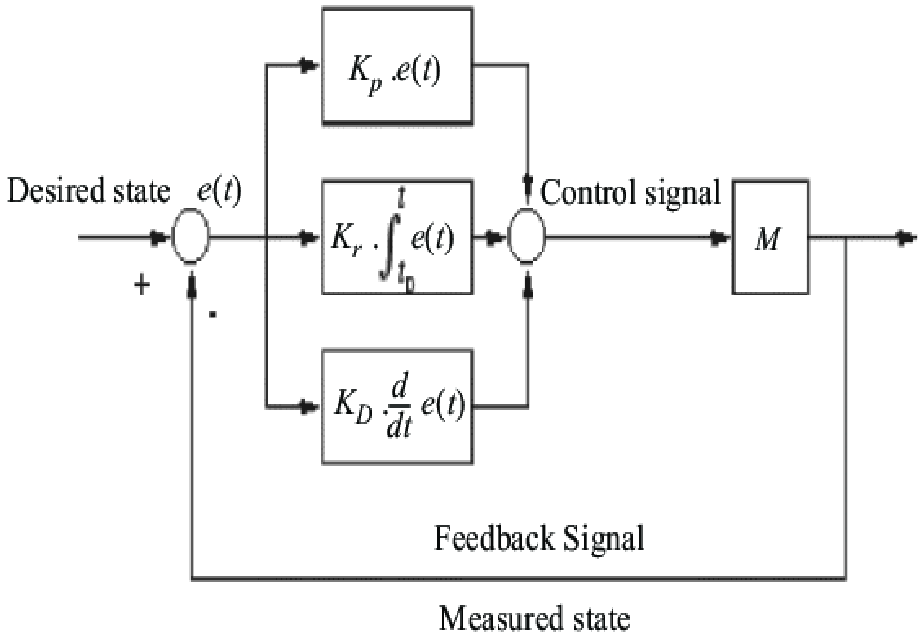
- proportional control action (where the control action is proportional to the actuating error signal, which is the difference between the input and the feedback signal),
- integral control action (where the control action is proportional to the integral of the actuating error signal), and
- derivative control action (where the control action is proportional to the derivative of the actuating error signal).

Where many plants are controlled directly by a single digital computer the majority of the control loops may be handled by PID control schemes.
The PID control action in analog controllers is given by

Formula for PID value:

$$K_p e + K_d \frac{de}{dt} + K_i \int_0^t e(t)\,dt$$

- **K_p : proportional gain**
- **K_d : derivative gain**
- **K_i : Integral gain**



Where *e(t)* is the input to the controller (the actuating error signal), *m(t)* is the output of the controller (the manipulating signal), *K* is the proportional gain, *TTii*is the integral time (or reset time), and *TTdd*is the derivative time (or rate time).

3.1 Tuning of gains:

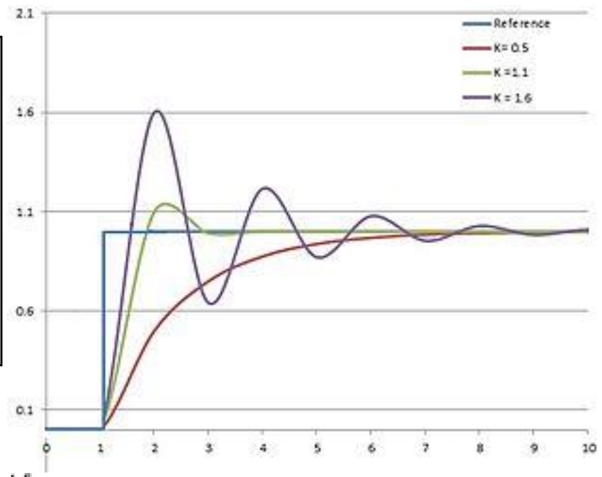
- To get appropriate values of 3 gains, we start with Kp, we set all the other two gains as zero and note down the oscillations. The best value of Kp will be the one which balances the ball as early as possible.

- Similar way we do the same with kd , for this part we set Ki and Kp as zero.

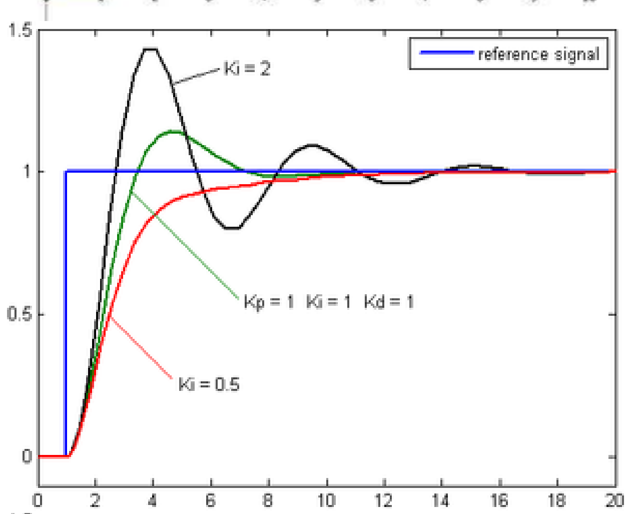
- And for Ki we set Kd and Kp as zero.
- The best values would be the one for which, the set point is reached early.

- The above method is called TRIAL AND ERROR method.

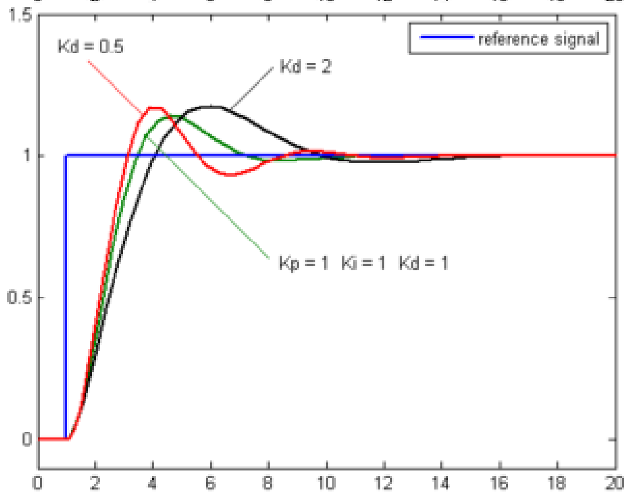
S
E
T
P
O
I
N
T



GRAPH OF Kp (proportional gain)



GRAPH OF Kp (proportional gain)

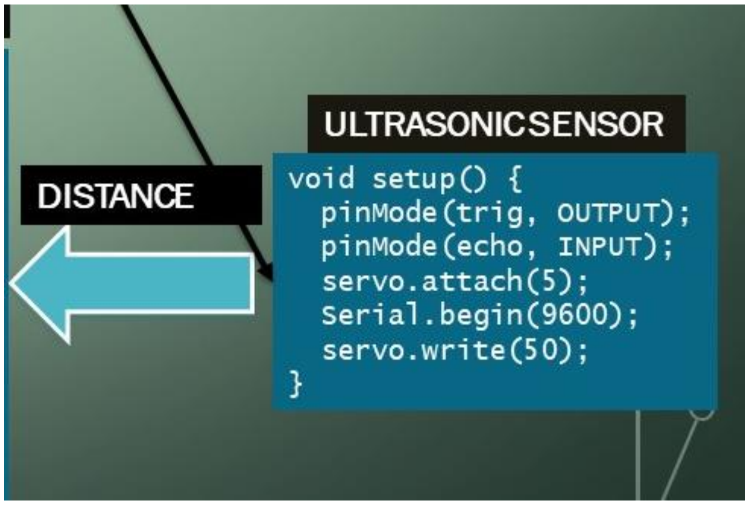


GRAPH OF Kp (proportional gain)

TIME IN SEC

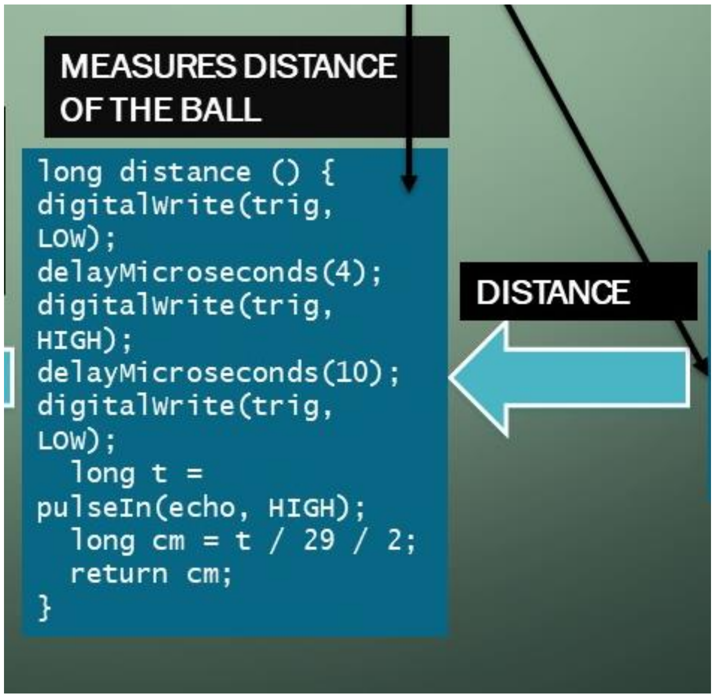
4.0 Code:

We used Arduino IDE for running our code and for uploading it into Arduino uno board.
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.



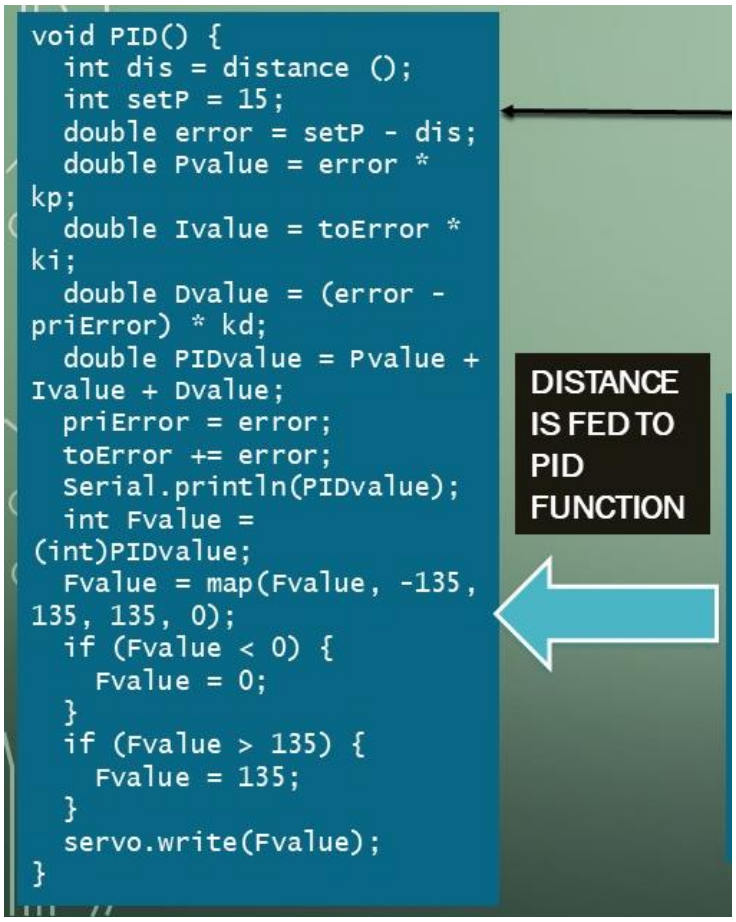
```
void setup() {
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  servo.attach(5);
  Serial.begin(9600);
  servo.write(50);
}
```

This part of the code ids for Ultrasonic sensor.



```
long distance () {
digitalWrite(trig,
LOW);
delayMicroseconds(4);
digitalWrite(trig,
HIGH);
delayMicroseconds(10);
digitalWrite(trig,
LOW);
  long t =
pulseIn(echo, HIGH);
  long cm = t / 29 / 2;
  return cm;
}
```

The function measures the distance of the ball from Ultrasonic sensor.



```
void PID() {
  int dis = distance ();
  int setP = 15;
  double error = setP - dis;
  double Pvalue = error *
kp;
  double Ivalue = toError *
ki;
  double Dvalue = (error -
priError) * kd;
  double PIDvalue = Pvalue +
Ivalue + Dvalue;
  priError = error;
  toError += error;
  Serial.println(PIDvalue);
  int Fvalue =
(int)PIDvalue;
  Fvalue = map(Fvalue, -135,
135, 135, 0);
  if (Fvalue < 0) {
    Fvalue = 0;
  }
  if (Fvalue > 135) {
    Fvalue = 135;
  }
  servo.write(Fvalue);
}
```

This is the PID function which calculates Kp, Ki, Kd values by finding the error.

Fvalue = map(Fvalue, -140, 140, 140, 0); // we use a map function to give a value to servo motor
 if (Fvalue < 0) {
   Fvalue = 0;
 }
 if (Fvalue > 140) {
   Fvalue = 140;
 }
 servo.write(Fvalue);   // This value is fed to servo motor which turns accordingly

4.0 Conclusion:

- PID control mechanism is uses different gains like kp, ki, kd to run the servo motor accordingly and balance it.
- The system balanced with very less percentage error of nearly 1.9%. It can be reduced to zero by using highly sensitive ultrasonic sensor and low friction assembly.
- Considering the static error, it is possible to argue that the main reason for not attaining a small enough error may not be the solely linear control but rather the motor specifications as aforementioned.
- A naturally unstable or semi-stable system is always in need of state adjustment to some extent. In the case of a ball balancing platform, the state to be adjusted is the platform inclination and it is actualized through servo motors. Thus, the adjustment has to be controlled. This can often be executed manually but within the field of control theory the actions are executed by the controller e.g. PID.
- The simulation we do in tinker cad has very less error percentage compare to the physical assembly. experimental limitations are inherently different than theoretical limitations due to time discretization, possibly limited computing power and non-ideal physical construction parameters.
- In this experiment, we balanced the ball on x-axis, but this can be extended to 3-dimension. By using multiple servo-motors, multiple Infrared sensors 3-D balance can be made.

Appendix

A.1 Error Analysis:

For error analysis, we have taken 10 readings. The set point is at a distance of 15cm from the ultrasonic sensor. We calculated the values of set point obtained when we ran the assembly.

Sno	Setpoint (Xo) in cm	Observed set point(X1) in cm	Error (|Xo – X1|) in cm	Percentage error in %
1	15	15.35	0.35	2.33
2	15	14.80	0.20	1.33
3	15	15.20	0.20	1.33
4	15	15.10	0.10	0.67
5	15	14.85	0.15	1.00
6	15	14.50	0.50	3.33
7	15	14.90	0.10	0.67
8	15	15.45	0.55	3.67
9	15	15.50	0.50	3.33
10	15	14.75	0.25	1.67

Total_error = 0.35+0.20+0.20+0.10+0.15+0.50+0.10+0.55+0.50+0.25
Average error = Total_error/10
Average error = 0.29
Percentage error = 0.29/15 = 1.933%
We have taken 10 readings to do the error analysis

REASONS FOR ERROR:

1. **Friction between the ball and the balancing part**
   - **in the above case, the ball might sometimes stop before the assumed set point**
2. **Accuracy of Ultrasonic sensor**
   - **Ultrasonic sensors are available in varying accuracy , we used a general sensor available in the market.**

A.2 References:

[1]. S. Bittanti, “James clerk maxwell, a precursor of system identification and control science,” *International Journal of Control*, vol. 88, no. 12, pp. 2427– 2432, 2015.

[2]. A. AG, “*Arduino uno rev3*.” https://store.arduino.cc/arduino-uno-rev3. Accessed: 2019-03-26.

[3]. Omega.uk website for more info on PID. Click here.

[4]. https://www.arduino.cc/en/software for compiling the code

[5]. PID control ball balance demo by youtuber. CLICKHERE.

[6]. ***The future of** PID control*
KJ Åström, T Hägglund - **Control** engineering practice, 2001 – Elsevier

[7]. “*Tuning rules for feedforward control from measurable disturbances combined with PID control”* JlGuzmán, THägglund - International **Journal** of **Control**, 2021 - Taylor & Francis

[8]. JonHub, “*A realtime digital signal processing (dsp) library for arduino*.” https://github.com/JonHub/Filters. Accessed: 09.04.2019.

Source code

```
#include <Servo.h>
Servo servo;

#define trig 2 // echo trig pins of ultrasonic sensor
#define echo 3 // 2 and 3 digital pins from arduino are connected

#define kp 10   // final value after tuning
#define ki 0.02 // final value after tuning
#define kd 3500 // final value after tuning

double priError = 0; // pri error means previous error
double toError = 0;  // toerror is total error

void setup()
{
    // The ultrasonic sensor Trig pin set as output pin and Echo pin
set as an input pin
    pinMode(trig, OUTPUT);
    pinMode(echo, INPUT);
    // The servo motor pin is set
    servo.attach(5);
    // The serial monitor is beginning
    Serial.begin(9600);
    // The servo motor rotates 50 degrees
    servo.write(50);
}
void loop()
{
    // Pid function
    PID();
    int a = distance();
    Serial.println(a);
    // prints the instantaneous distance of ball on serial monitor
}
/*This function measures distance by trigerring servo motor*/
long distance()
{
    digitalWrite(trig, LOW);
    delayMicroseconds(4);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    long t = pulseIn(echo, HIGH);
    long cm = t / 29 / 2;
    return cm;
}
void PID()
{
    int dis = distance();

    int setP = 15;
    double error = setP - dis;
    // setpoint is taken as 15cm
    // dis is distance from ultrasonic sensor

    /****************************************/
    double Pvalue = error * kp;
    double Ivalue = toError * ki;
    double Dvalue = (error - priError) * kd;
    /****************************************/

    double PIDvalue = Pvalue + Ivalue + Dvalue;
    priError = error;
    toError += error;
    // toError signifies total error
    Serial.println(PIDvalue);
    // Prints the PID value on serial monitor
    int Fvalue = (int)PIDvalue;
    // Fvalue will be the PID value as an intger
    // Thsi integer value is used for tuning purpose

    Fvalue = map(Fvalue, -140, 140, 140, 0);
    // fro mapping, map function is used

    if (Fvalue < 0)
    {
        Fvalue = 0;
    }
    if (Fvalue > 140)
    {
        Fvalue = 140;
    }

    // Final F value is sent to servo moter to rotate accordingly
    servo.write(Fvalue);
}
```

============END OF REPORT=============

©© 2022 DIY Group-6 sec-12