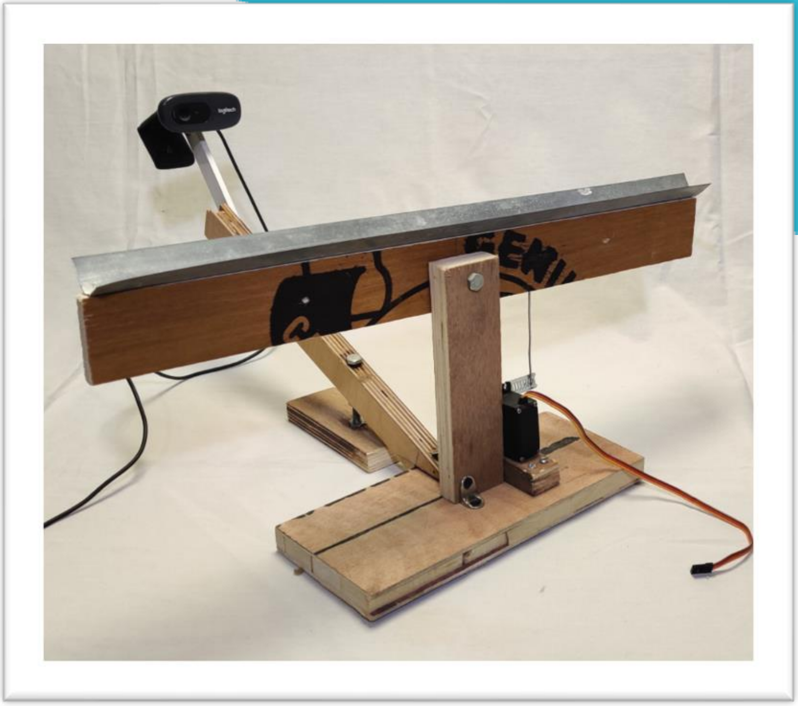
PID Ball Balancer



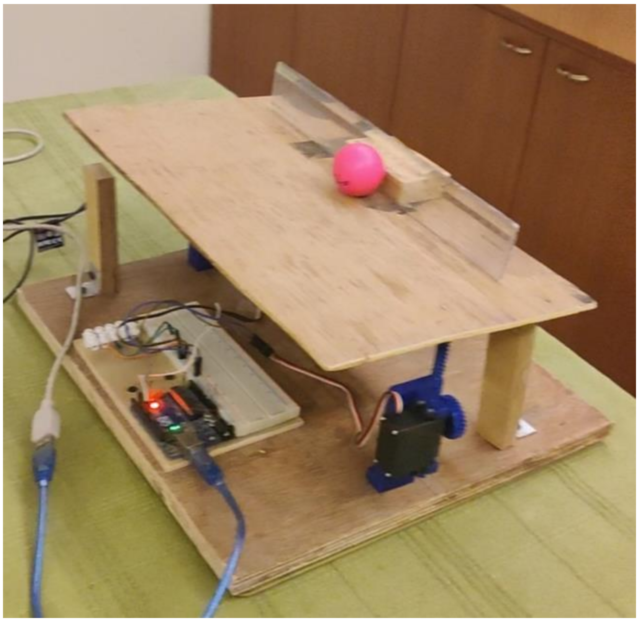
mihir.vardhan2002@gmail.com mihirvardhan.com July, 2019

OBJECTIVE

I have always been fascinated by control algorithms. I was first exposed to this seemingly simple concept in my preparations for the Indian Robot Olympiad. We wanted to build a gyro stabilized frame that would stay in place no matter the orientation of the robot. At the time we thought it was going to be an easy linear algorithm. Little did we know, we would be introduced to one of the most fascinating concepts of robotics – control algorithms.

Despite implementing our code for the competition, I wanted a proper, in depth understanding of control algorithms – namely the PID algorithm.

I love to learn things hands-on and so, I built the 'PID Ball Balancer.' This robot is designed to keep a ball balanced on a strip of aluminum. Creating an interesting demonstration for kids, the robot manages to teach a seemingly difficult concept with its practical application.



The first version of the balancer. Not as elegant but served as a proof of concept

THEORY

A control system is one which regulates the behavior of a system output based on an external input. Control systems can be used for heating or cooling a house or controlling large industrial systems. A feedback control system compares the process variable (value being controlled) with the desired setpoint. Using this comparison it produces a control signal to bring the process variable to the set point.

Though the above paragraph may seem complicated, comparing it to our ball balancer, we can understand it easily. The position of the ball is the process variable – the value we must control – and the middle of the aluminum strip is the desired setpoint. The control signal is then sent to a servo motor to adjust the angle of the aluminum strip and bring the ball back to the center.

This simple example can be used to explain all applications of control systems!

PID CONTROL

This project demonstrates the 'Proportional, Integral and Derivative' controller, commonly known as PID.

The most intuitive control algorithm is Proportional control. Take the ball balancer for example. The farther out the ball moves, the greater should be the angle in the opposing direction. As the ball comes closer and closer the angle reduces and the ball should ideally come to a stop right in the middle. However, this is not the case. Over the next few paragraphs I will discuss how 'Integral' and 'Derivative' control is also used in unison with 'Proportional' control to make an accurate control system.

The two biggest shortcomings of a Proportional control system is that the system can accumulate errors for long periods of time – making the system overdamped. Or, the system can oscillate

around its set point, making it underdamped.

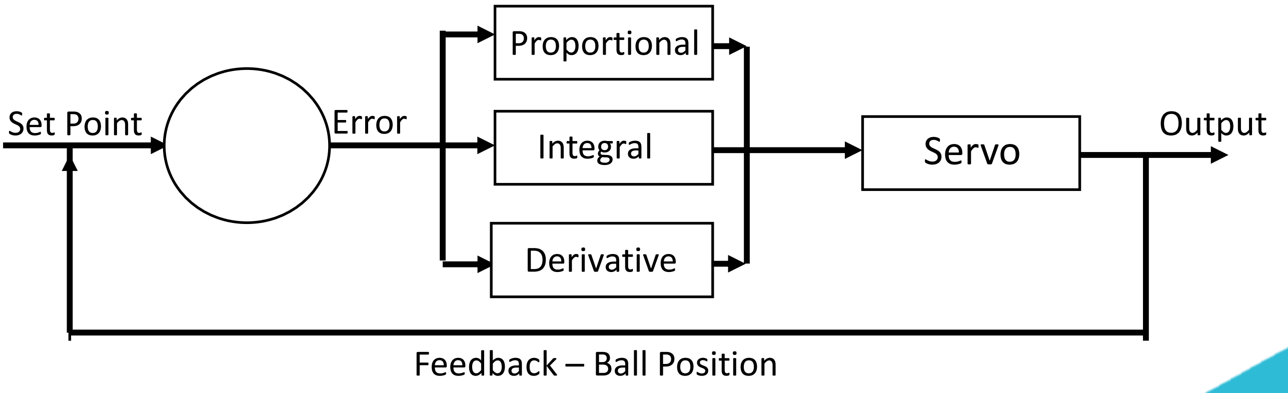
Derivative action – In a PID loop, the derivative part measures the rate of change of error with time. In the ball example, if the measured distance from the center point approaches rapidly, the servo reduces it angle quickly to help the ball come to a timely stop. If the measured distance increases rapidly away from the center point, the angle is increased rapidly to help the ball come to a stop.

At this point, you might think that with the 'Proportional' and 'Derivative' control working in unison, the problem of underdamped and overdamped is removed. However, this is another form of error that comes into play known as 'Steady State Error.' This is tackled by the integral action of the controller.

Integral action – Let us assume our balancing aluminum strip is 1m long. If the ball is, say, 2cm away from the center position, the proportional control does not have a significant enough effect on the motor's angle and the velocity of the ball will be nearly 0, rendering the derivative control ineffective. This is known as steady state error.

The integral control sums up the error in position over time applying an ever-increasing effort to make the ball reach the center position and reduce the error to 0.

BLOCK DIAGRAM



BUILD OVERVIEW

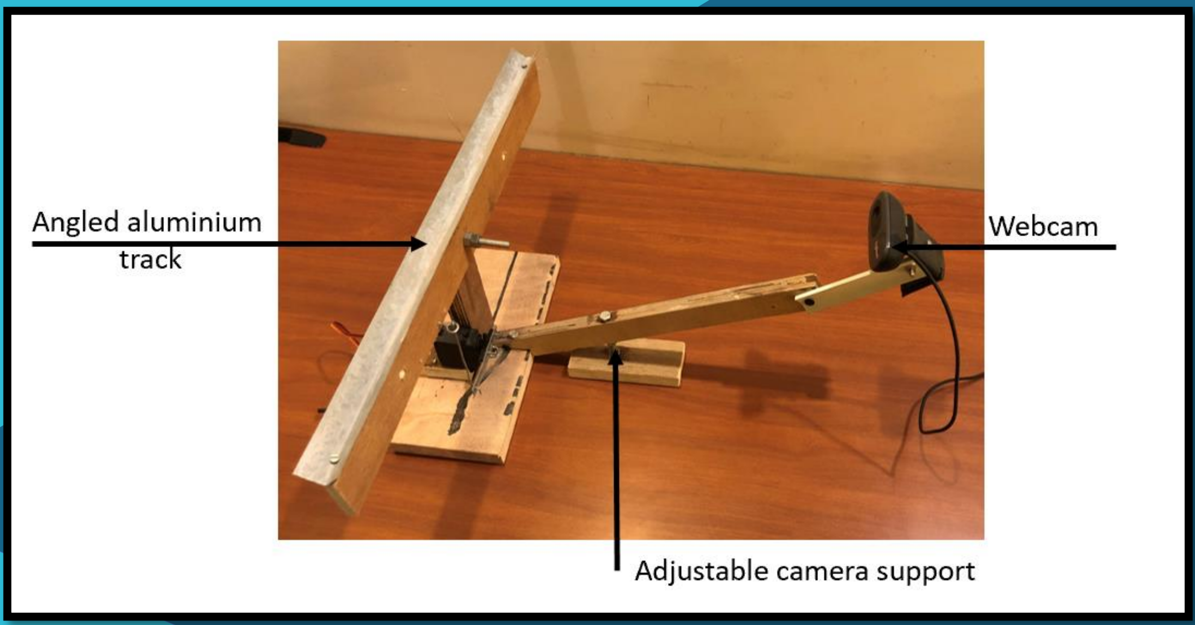
The setup contains a servo motor to control the angle of an aluminum strip using a connecting rod. A processing program uses a webcam to track the ball by looking for its color and sends the x axis coordinate to the Arduino over the COM port. The Arduino maps this X coordinate to a servo motor value.

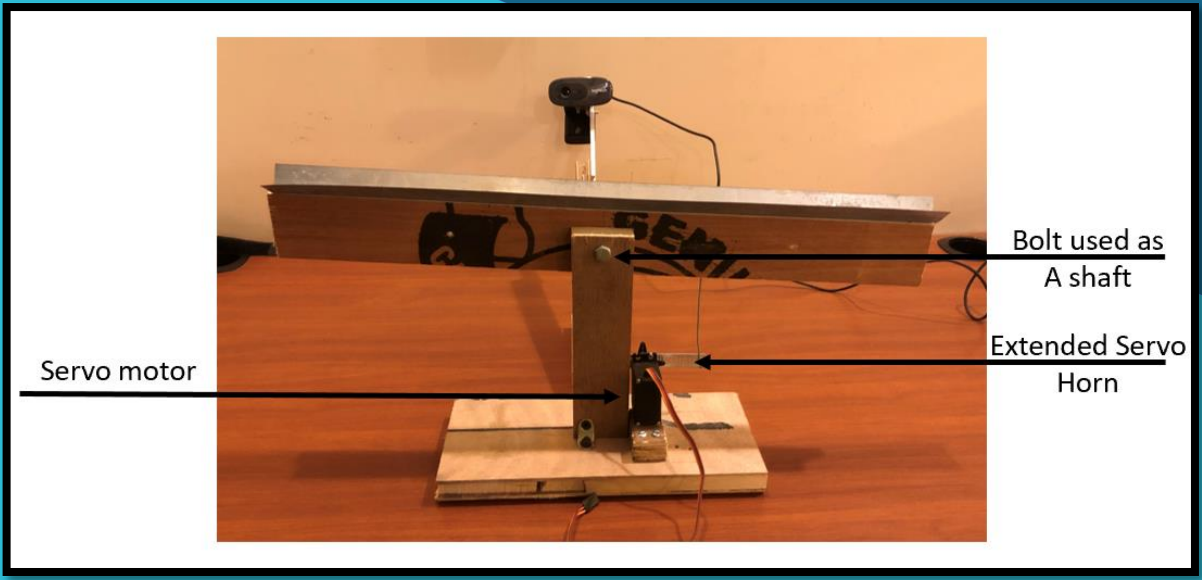
BILL OF MATERIALS

As with most of my projects, all the wood for this project is recycled and the aluminum channel is from an old window.

- 1x MG995 Servo Motor
- 1x Arduino Uno
- 1x Webcam
- Assorted pieces of scrap wood
- L-Bracket
- Long angled aluminum channel
- GI wire shaft
- 1x Quarter inch bolt as a shaft
- 2x Quarter inch nut as a lock nut

ROBOT OVERVIEW





RESOURCE LINKS

-https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-06-principles-of-automatic-control-fall-2012/lecture-notes/MIT16_06F12_Lecture_10.pdf

-https://web.stanford.edu/class/archive/ee/ee392m/ee392m.1034/Lecture4_PID.pdf

-https://www.omega.co.uk/prodinfo/pid-controllers.html

-https://www.ni.com/en-us/innovations/white-papers/06/pid-theory-explained.html