

RELATÓRIO DO EXPERIMENTO

Ballance Ball

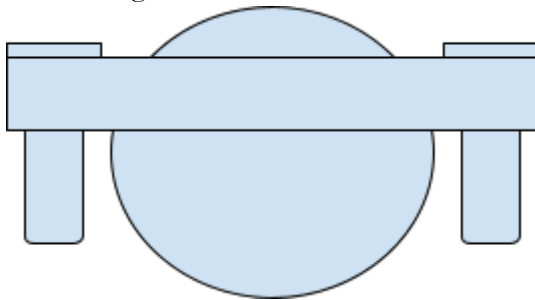
CAMPINA GRANDE
1 de MARÇO de 2024
Wagner Edney dos Santos Rodrigues Filho

1. INTRODUÇÃO

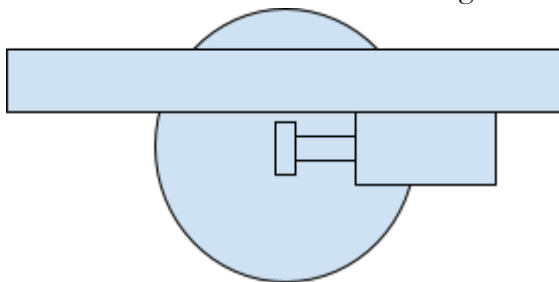
O objetivo desta missão é implementar e projetar um sistema que utilize arduino para equilibrar uma carga suspensa por uma esfera. Utilizando um sistema PID com sensores ultra sônicos para entrada de dados e servomotores como saída de dados.

2. MONTAGEM

A montagem é feita da forma demonstrada na imagem abaixo



Usando 2 conjuntos de sensores e servomotores de funcionamento contínuo, um para cada eixo de movimento da esfera. Conectado a uma roda de borracha, para maior aderência, o servomotor será montado da forma da imagem abaixo.



Com o sensor apontado para baixo, o código foi feito usando o princípio de PID para que conforme a mesa “se” distancia do sensor, o motor irá se mover para rebalancear o quadrado. O código utilizado será disponível a baixo:

```
#include <Servo.h>

#define TRIGGER_PIN 2
#define ECHO_PIN 3
#define SERVO_PIN 9

Servo servo;

void setup() {
  Serial.begin(9600);
  pinMode(TRIGGER_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  servo.attach(SERVO_PIN);
}

void loop() {
  long duration, distance;
```

```

// Limpa o pino de trigger
digitalWrite(TRIGGER_PIN, LOW);
delayMicroseconds(2);

// Envia um pulso de 10 microssegundos para o pino de trigger
digitalWrite(TRIGGER_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIGGER_PIN, LOW);

// Lê o tempo de retorno do eco
duration = pulseIn(ECHO_PIN, HIGH);

// Calcula a distância em centímetros
distance = duration * 0.034 / 2;

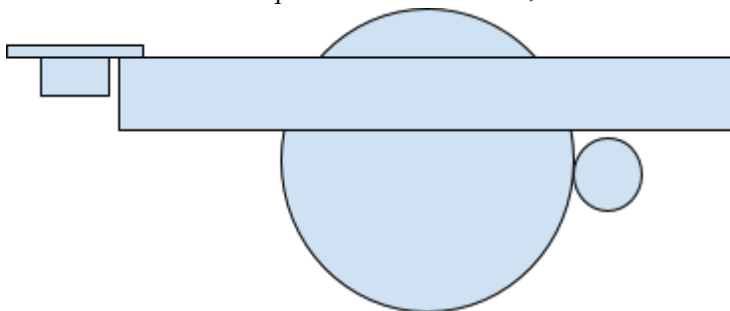
Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");

// Verifica se a distância é menor ou igual a 10cm
if (distance <= 10) {
    // Define a velocidade do servo motor contínuo para girar no sentido horário
    int speed = map(distance, 0, 10, 1000, 2000); // Mapeia a distância para a velocidade
(1000 a 2000 microssegundos)
    servo.writeMicroseconds(speed);
} else {
    // Define a velocidade do servo motor contínuo para girar no sentido anti-horário
    int speed = map(distance, 10, 300, 2000, 1000); // Mapeia a distância para a velocidade
(2000 a 1000 microssegundos)
    servo.writeMicroseconds(speed);
}

delay(100); // Espera 100 milissegundos antes de fazer a próxima leitura
}

```

Para manter o quadrado balanceado, o sensor será disposto como indica a imagem abaixo



3. CONCLUSÃO

Com o projeto feito, pode concluir as claras vantagens que se tem ao se utilizar um sistema PID em comparação a um sistema “não adaptativo”, como capacidade de

lidar com uma ou mais variáveis. Com isso, máquinas projetadas usando esse princípio podem ser mais flexíveis e modulares, algo muito desejado na área da automação, por exemplo. Além disso, isso me fez perceber a facilidade e praticidade de se trabalhar com um microcontrolador como o arduino, relativamente acessível, simples de se usar e principalmente open source.

4, REFERÊNCIAS

- <https://www.tinkercad.com/things/5snktGTw7p1-codigo-sensor-motor/editel?returnTo=%2Fdashboard>
- <https://docs.arduino.cc/hardware/uno-rev3>
- <https://www.rugged-circuits.com/10-ways-to-destroy-an-arduino/>
- <https://docs.arduino.cc/learn/electronics/servo-motors>
- <https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/>
- <https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>
- <https://www.youtube.com/watch?v=kUHmYKWwuWs>
- <https://www.youtube.com/watch?v=1WnGv-DPexc>
- <https://www.youtube.com/watch?v=wkfEZmsQqiA>
- <https://www.youtube.com/watch?v=UR0hOmjaHp0>
- <http://www.lamja.com/?p=504>