



UNIVERSIDADE FEDERAL DE LAVRAS

JOÃO SOUZA SANTIAGO - 202111159
THIAGO HENRIQUE REZENDE - 202110845
PEDRO HENRIQUE GUIMARÃES CARVALHO - 202110328

Modelagem de Sistemas Dinâmicos

Trabalho prático

Controle de balanço

LAVRAS

2023

SUMÁRIO

1 OBJETIVO.....
2 MATERIAIS UTILIZADOS.....
3 DESENVOLVIMENTO DO MODELO
4 SISTEMA EM MALHA FECHADA.....
5 CÓDIGO DESENVOLVIDO.....
6 VÍDEO DO FUNCIONAMENTO.....
7 DIFICULDADES ENFRENTADAS.....
8 CONCLUSÃO.....
9 REFERÊNCIAS.....

1. Objetivos

O objetivo deste trabalho é estudar e analisar um modelo de balanço com controle PID (Proporcional, Integral e Derivativo) com a finalidade de desenvolver uma estratégia de controle que permita manter uma bola em equilíbrio no ponto médio de uma superfície inclinada. A modelagem e simulação serão realizadas para entender o comportamento dinâmico do sistema e projetar um controlador PID eficiente, capaz de manter a bola na posição central, apesar das perturbações externas.

2. Materiais utilizados

Durante o desenvolvimento do projeto de controle de balanço, foram utilizados os seguintes materiais:

- Arduino UNO;
- Servo motor;
- Sensor ultrassônico;
- Estrutura feita em impressão 3d;
- Base rígida de madeira;
- Bola de ping pong;

Alguns dos recursos computacionais foram utilizados também:

- Arduino IDE (Programação do Arduino);
- Matlab (Simulação e estudo do sistema);
- Fusion 360 (Modelagem mecânica da estrutura);

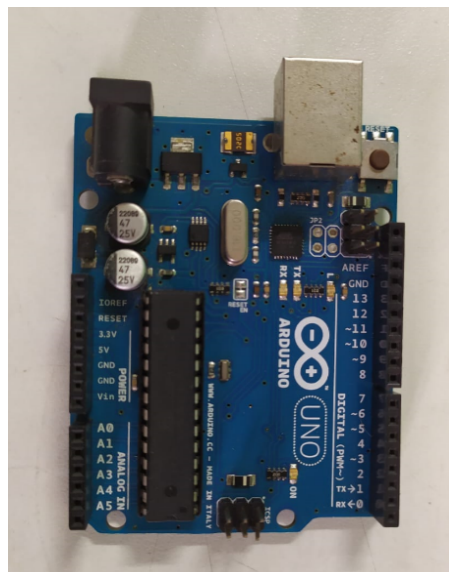


Figura 1 - Arduino UNO

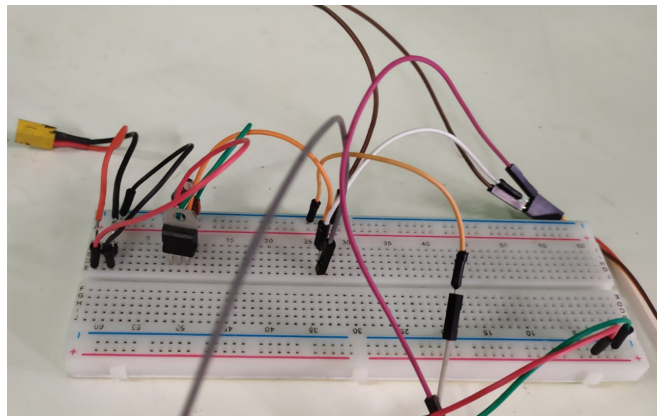


Figura 2 - Protoboard com xt30 e lm7805

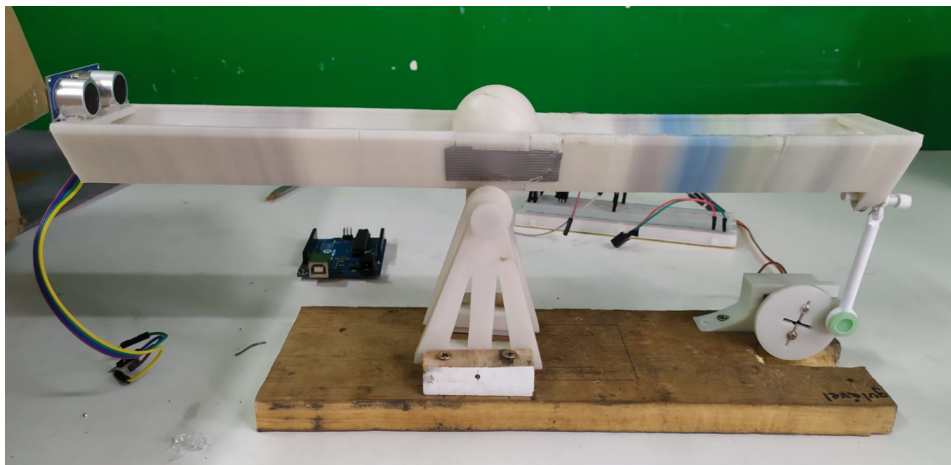


Figura 3 - Projeto impresso com os sensores

3. Desenvolvimento do modelo

Primeiramente, para desenvolver o modelo desse sistema analisamos o sistema físico dele, como é mostrado na figura anexada na próxima página:

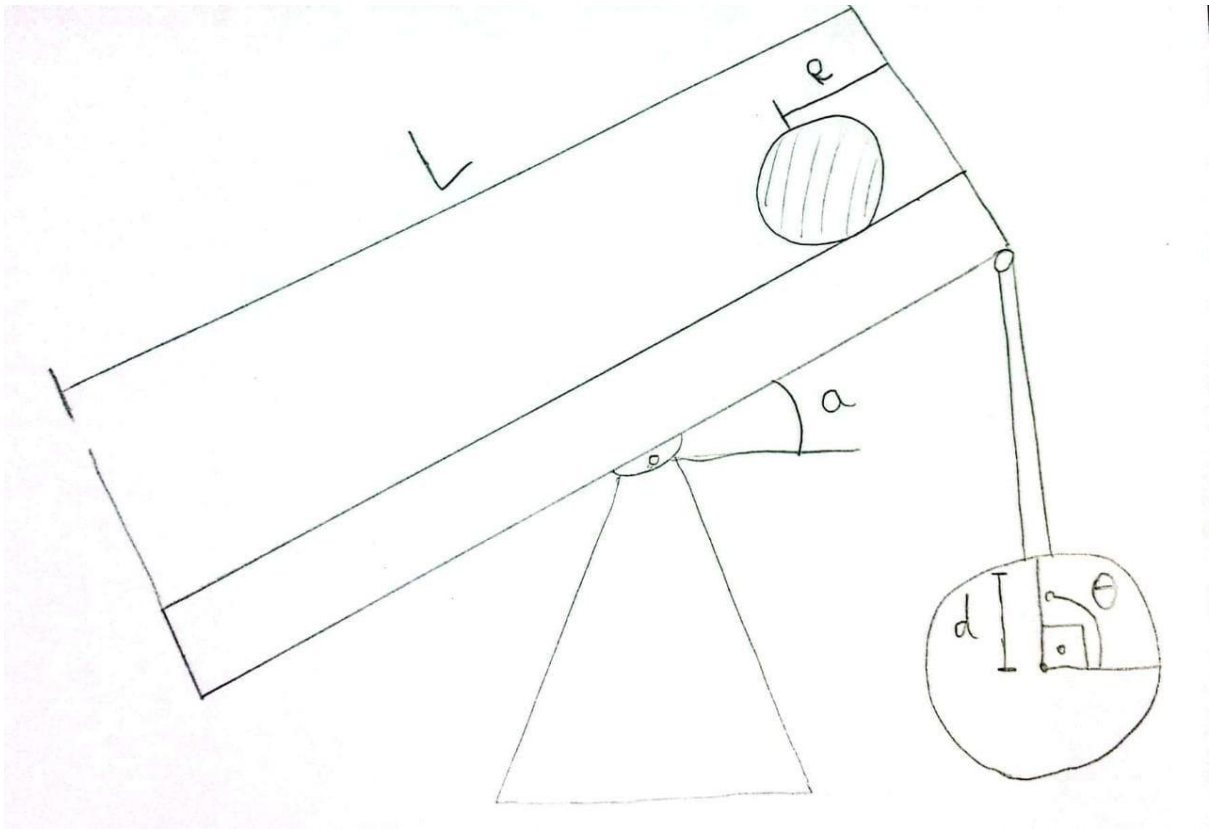


Figura 4 - Sistema físico

Segundamente, após a análise física do sistema relacionamos ele com os conhecimentos adquiridos em sala de aula para escolher o melhor método que descreve a equação do movimento da bola. E após relacionar essas questões, optamos por utilizar o método de Lagrange, pois ele é mais útil para sistemas com um número maior de graus de liberdade ou com coordenadas generalizadas que o método de Newton, tornando o processo de análise e obtenção das equações de movimento mais simples.

Nesse sentido, obtivemos pelo método de Lagrange a seguinte equação que rege o movimento da bola em nosso sistema:

$$0 = \left(\frac{J}{R^2} + m \right) \ddot{r} + mg \sin \alpha - m r \dot{\alpha}^2$$

$$a = \frac{d}{L} \cdot \theta$$

Figura 5 - Equação lagrangeana para o movimento da bola

Terceiramente, analisamos como essa equação se comporta simulando-a no recurso “Simulink” do Matlab, e definimos alguns valores para essa simulação, sendo eles:

- m (massa da bolinha) = 0,0025 kg;
- g (gravidade) = -9,8 m/s²;
- l (comprimento) = 0,3 m;
- d (distância servo e haste) = 0,02 m;
- J (inércia) = $5,4 \times 10^{-7}$ kg*m²;
- r (distância) = 0,015 m;

A partir disso, montamos o seguinte diagrama no Simulink:

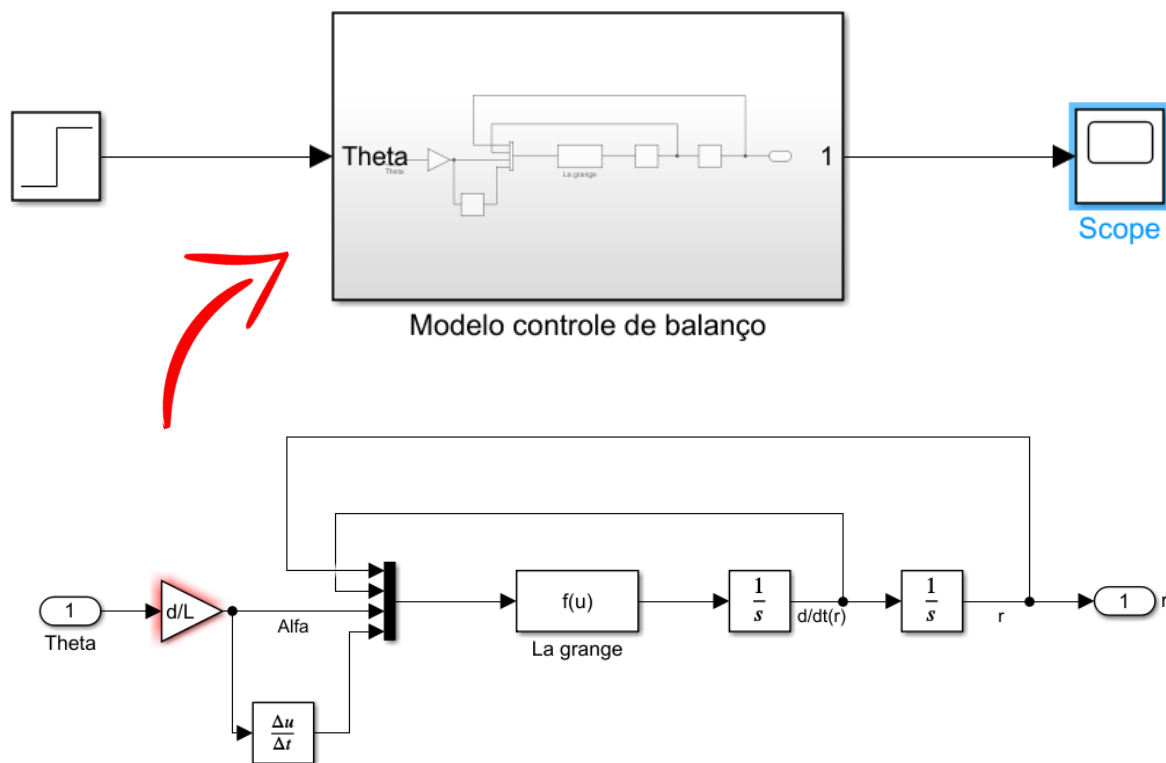


Figura 6 - Diagrama Simulink

Logo, obtivemos a seguinte resposta ao degrau:

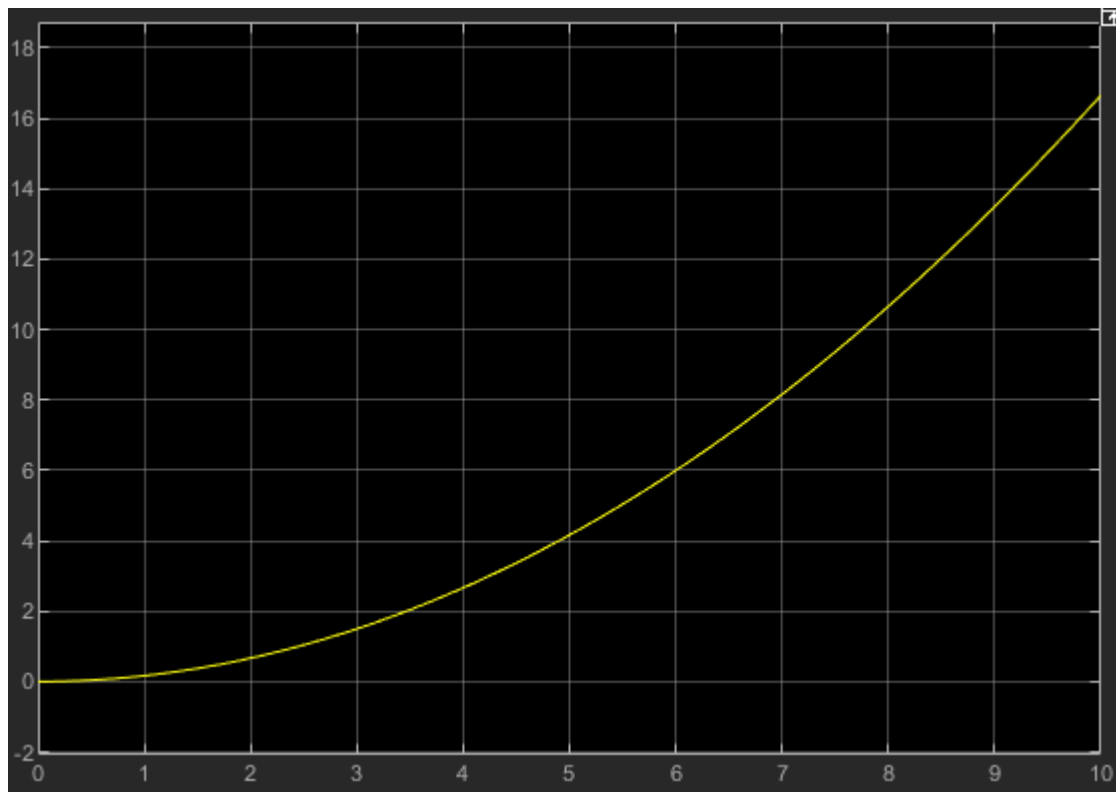


Figura 7 - Resposta ao degrau

A partir desse gráfico, fica claro que o sistema é instável em malha aberta, fazendo com que a bola role diretamente para fora da extremidade da viga. Portanto, algum método de controle da posição da bola nesse sistema é necessário.

O próximo passo foi achar a função de transferência do sistema, e para isso seguimos os seguintes passos:

$$0 = \left(\frac{J}{R^2} + m \right) \ddot{R} + mg \sin \alpha - mR\dot{\alpha}^2.$$

Linearizando em torno de $\alpha = 0$.

$$\Rightarrow \left(\frac{J}{R^2} + m \right) \ddot{R} = -mg\alpha$$

$$\alpha = \frac{d}{L} \cdot \theta \rightarrow \left(\frac{J}{R^2} + m \right) \ddot{R} = -mg \frac{d}{L} \cdot \theta$$

$$\left(\frac{J}{R^2} + m \right) \cdot R(s) \cdot s^2 = -mg \frac{d}{L} \cdot \theta(s)$$

$$\Rightarrow \frac{R(s)}{\theta(s)} = - \frac{mgd}{L \left(\frac{J}{R^2} + m \right)} \cdot \frac{1}{s^2} \quad \left(\frac{\text{m}}{\text{rad}} \right)$$

$$\Rightarrow G_2(s) = \frac{4,9 \times 10^{-11}}{0,00147 \cdot s^2} = \frac{0,3}{s^2} \quad \frac{\text{m}}{\text{rad}}$$

Figura 8 - Função de transferência

Para melhor entendimento e aprofundamento do estudo, também representamos o sistema no espaço de estados:

$$\begin{bmatrix} \dot{R} \\ \ddot{R} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} R \\ \dot{R} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{mgd}{L(\frac{3}{2}R^2 + m)} \end{bmatrix} \theta$$

Figura 9 - Espaço de estados

Por fim, realizamos algumas simulações no MatLab para entender a resposta do nosso sistema de acordo com diferentes excitações, como é mostrado nas próximas páginas:

- Resposta ao degrau ($K_p = 1$):

```
clear;
clc;
close all;

m = 0.0025;
R = 0.015;
g = -9.8;
L = 0.3 ;
d = 0.02;
J = 5.4e-7;

s = tf('s');
P_ball = -m*g*d/L / (J/R^2+m) / s^2;

Kp = 1;
C = pid(Kp);
sys_cl=feedback(C*P_ball,1);

step(sys_cl);
axis([0 50 0 2]);
```

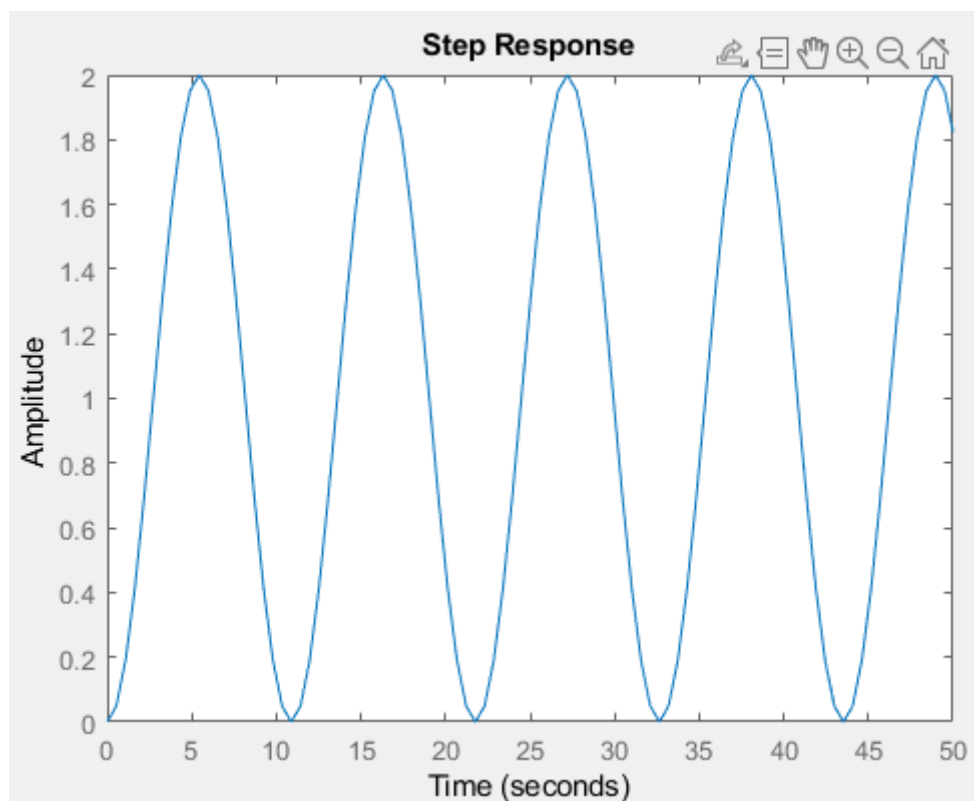


Figura 10 - Resposta ao degrau

- Resposta ao degrau ($K_d = 10$, $K_p = 10$, $K_i = 0$):

```
clear;
clc;
close all;

m = 0.0025;
R = 0.015;
g = -9.8;
L = 0.3 ;
d = 0.02;
J = 5.4e-7;

s = tf('s');
P_ball = -m*g*d/L/(J/R^2+m)/s^2;

Kp = 10;
Kd = 10;
Ki = 0;

C = pid(Kp,Ki,Kd);

sys_cl=feedback(C*P_ball,1);

t=0:0.01:5;
step(sys_cl)
```

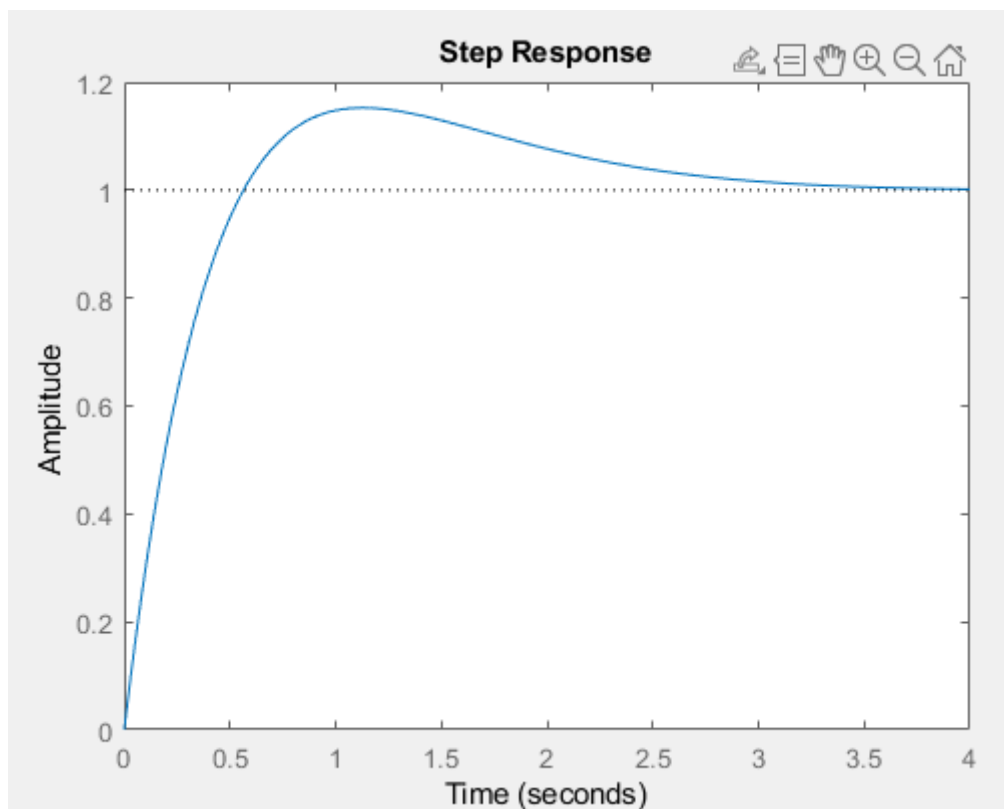


Figura 11 - Resposta ao degrau

- Resposta ao degrau ($K_d = 10$, $K_p = 10$, $K_i = 10$):

```
clear;
clc;
close all;

m = 0.0025;
R = 0.015;
g = -9.8;
L = 0.3 ;
d = 0.02;
J = 5.4e-7;

s = tf('s');
P_ball = -m*g*d/L / (J/R^2+m) / s^2;

Kp = 10;
Kd = 10;
Ki = 10;

C = pid(Kp,Ki,Kd);

sys_cl=feedback(C*P_ball,1);

t=0:0.01:5;
step(sys_cl)
```

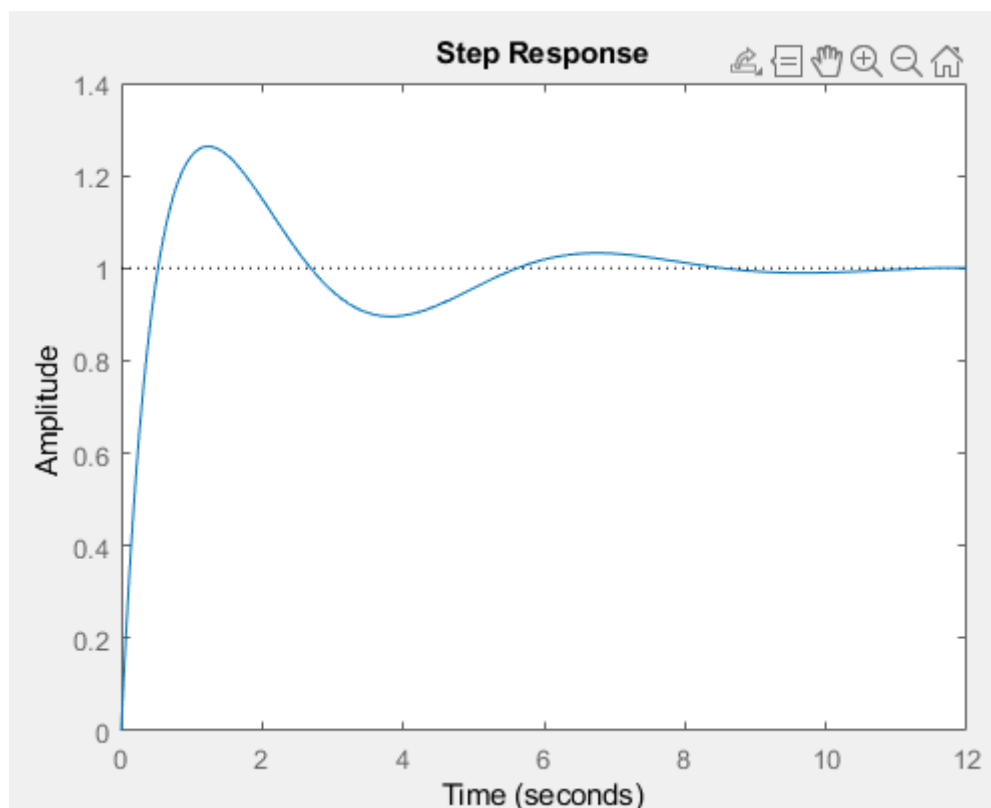


Figura 12 - Resposta ao degrau

- Resposta ao degrau ($K_d = 40$, $K_p = 30$, $K_i = 5$):

```
clear;
clc;
close all;

m = 0.0025;
R = 0.015;
g = -9.8;
L = 0.3 ;
d = 0.02;
J = 5.4e-7;

s = tf('s');
P_ball = -m*g*d/L/(J/R^2+m)/s^2;

Kp = 30;
Kd = 45;
Ki = 5;

C = pid(Kp,Ki,Kd);

sys_cl=feedback(C*P_ball,1);

t=0:0.01:5;
step(sys_cl)
```

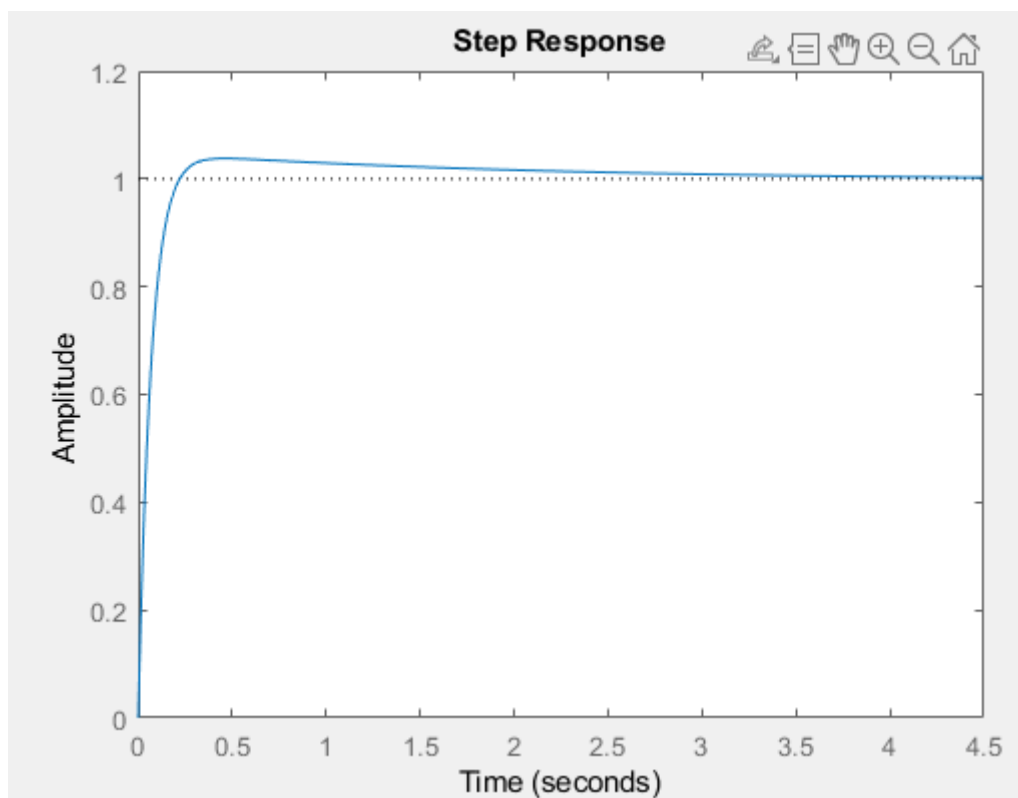


Figura 13 - Resposta ao degrau

Devido à essas simulações e aos testes realizados no projeto decidimos colocar os valores de PID da seguinte forma:

$K_p = 30;$

$K_d = 45;$

$K_i = 5;$

Pois essa combinação de valores foi a que melhor se adequa ao nosso sistema.

4. Sistema em malha fechada

Nosso sistema de controle pode ser representado pelo seguinte diagrama:

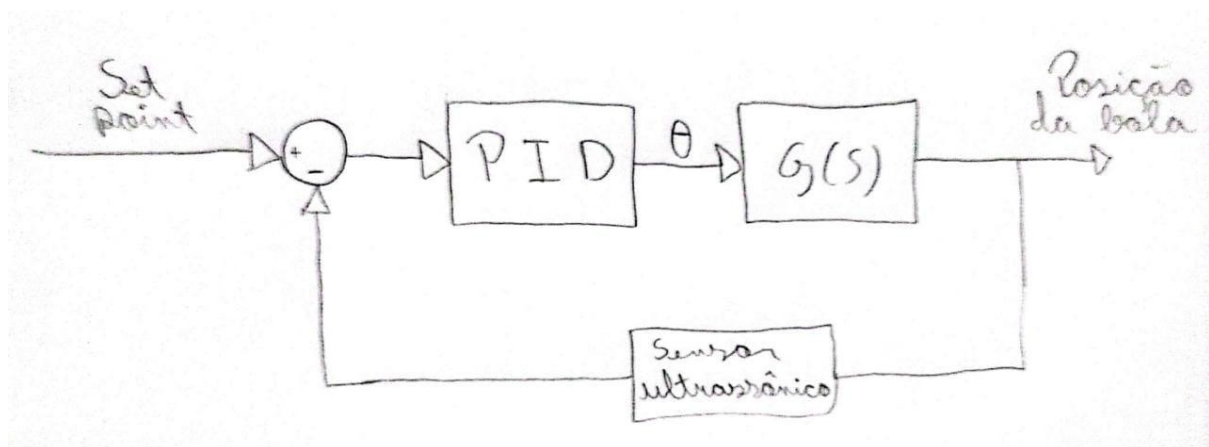


Figura 14 - Diagrama do sistema

5. Código desenvolvido

Para o funcionamento do projeto desenvolvemos o seguinte código para controle através do arduino:

```

#include <Wire.h>
#include <Servo.h>
#include <HCSR04.h>

Servo myservo;

#define TRIGGER 11
#define ECHO 12
UltrasonicDistanceSensor distanceSensor(TRIGGER, ECHO);

int Read = 0;
float distance = 0.0;
float elapsedTime, time, timePrev;
float distance_previous_error, distance_error;
int period = 50;

float kp=30;
float ki=3;
float kd=50;
float distance_setpoint = 15;
float PID_p, PID_i, PID_d, PID_total;

const double a=0.5;
long cm1=13;
int cmOld = 0;

void setup() {
  Serial.begin(9600);
  myservo.attach(9);
  myservo.write(125);
  time = millis();
}

```

Figura 15 - Código parte 1

```

void loop() {
  if (millis() > time + period)
  {
    time = millis();
    distance = readPosition();
    distance_error = distance_setpoint - distance;
    PID_p = kp * distance_error;
    float dist_difference = distance_error - distance_previous_error;
    PID_d = kd*((distance_error - distance_previous_error)/period);

    if(-3 < distance_error && distance_error < 3)
    {
      PID_i = PID_i + (ki * distance_error);
    }
    else
    {
      PID_i = 0;
    }

    PID_total = PID_p + PID_i + PID_d;
    PID_total = map(PID_total, -150, 150, 0, 150);

    if(PID_total < 20){PID_total = 20;}
    if(PID_total > 160) {PID_total = 160; }

    myservo.write(PID_total+30);
    distance_previous_error = distance_error;
  }
}

```

Figura 16 - Código parte 2


```

// Leitura da posição da bola + filtro

float readPosition() {
    delay(40);

    long duration, cm, cmn;
    unsigned long now = millis();
    int distancia = 0;

    distancia = distanceSensor.measureDistanceCm();

    cm = distancia;

    if(cm > 26)
    {cm=32;}

    cmn = a * cm + (1 - a) * cm1;
    Serial.print("Distancia: ");
    Serial.print(cm); Serial.print(" -- filtro --> ");
    Serial.println(cmn); //cmn is filtered value
    Serial.println("");
    delay(10);
    cm1 = (cmn + cmOld)/2;
    cmOld = cmn;
    Serial.print("FILTRO =");
    Serial.println(cmn);
    return (cmn);
}

```

Figura 17 - Código parte 3

6. Vídeo do funcionamento

O funcionamento prático do nosso projeto pode ser comprovado através do seguinte vídeo publicado no Youtube:

- <https://youtu.be/aqdYYsWxAYg>

7. Dificuldades enfrentadas

Durante o desenvolvimento do projeto de controle de balanço, encontramos algumas dificuldades que impactaram o progresso do trabalho. As principais dificuldades enfrentadas foram as seguintes:

7.1. Queima do Arduino

Uma das dificuldades encontradas foi a queima do Arduino devido à corrente excessiva demandada pelo servo motor utilizado no sistema. O alto consumo de corrente pelo servo motor excedeu a capacidade de fornecimento do Arduino, resultando em sua queima. Para solucionar esse problema, foi necessário utilizar uma fonte de alimentação externa, e escolhemos uma bateria Li-po de 2 células. Porém, a tensão fornecida por essa bateria é equivalente a 8,4V, e para que não haja problema no Arduino, dimensionamos um regulador de tensão com saída de 5V, assim o problema foi resolvido.

7.2. Limitação do servo motor

Outra dificuldade enfrentada foi a limitação do servo motor em suportar o levantamento repetitivo da estrutura. O servo motor utilizado não possuía torque suficiente para sustentar a estrutura de forma consistente, levando a oscilações e perda de estabilidade. Para contornar essa limitação, consideramos a substituição do servo motor por um modelo mais robusto e capaz de lidar com as exigências do sistema de balanço.

7.3 Sensor ultrassônico

O nosso sensor ultrassônico estava com problemas em detectar a bola em distâncias maiores que 20cm, causando certa instabilidade no sistema, além de em certos momentos captar outros objetos a não ser a bola. Para resolver esses problemas criamos um filtro em nosso código para aumentar a precisão da distância medida, porém, apesar de ter melhorado bastante, ainda não conseguimos eliminar todas as falhas, e só seria possível utilizando um sensor melhor.

8. Conclusão

Ao final deste relatório, podemos afirmar que, apesar das dificuldades enfrentadas, obtivemos sucesso no controle da posição da bola em um balanço utilizando um sistema de controle PID. Com base nos conhecimentos adquiridos em Modelagem de Sistemas Dinâmicos, foi possível projetar e implementar um controlador PID eficiente, que permitiu ao servo motor mudar a angulação da estrutura de forma adequada para controlar a posição da bola.

Nesse sentido, mesmo o projeto tendo algumas limitações, conseguimos utilizar nosso conhecimento teórico e prático adquiridos ao longo do semestre para conhecer nosso modelo e manipular o sistema a fim de controlar a posição da bola.

9. Referências

ELECTRONOBS. Arduino Tutorial 100: Arduino Wireless Communication HC-12. Disponível em: http://electronoobs.com/eng_arduino_tut100.php. Acesso em: (16/05/2023).

SHASHINY, Kiran. PidController_BallBalancingRobot. Disponível em: https://github.com/kiranshashiny/PidController_BallBalancingRobot. Acesso em: (16/05/2023).

NISE, N. S. Engenharia de sistemas de controle. 6 ed. São Paulo: Editora LTC, 2012.

OGATA, Katsuhiko. Engenharia de controle moderno. 5. ed. São Paulo: Prentice Hall, 2010.