



## Controle de posição vertical do sistema bola-cooler-tubo

Laboratório Integrador  
Professor: Daniel Furtado Leite

Luiz Fernando Alves Rodrigues - 201610292 - 22D  
Saulo de Freitas David - 201520775 - 22C

Lavras-MG  
Abril de 2022

# 1 Introdução

O seguinte projeto consiste em sustentar uma esfera em um tubo utilizando um cooler para gerar um fluxo de ar que mantém a esfera estabilizada em qualquer posição estabelecida pelo usuário levando em consideração o fluxo de entrada de ar e reajustando a tensão de forma adequada para sustentar a bola.

Para a execução do projeto foram utilizados a plataforma arduino em conjunto com o software MatLab, e um controlador criado a partir dos dados coletados pelos mesmos.

## 2 Materiais

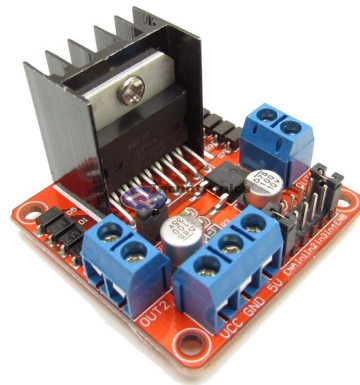
Para a criação física do projeto foram utilizados um tubo plástico ligado a uma carcaça de fonte de computador, onde foram criadas aberturas para entrada de ar e para a inserção do cooler alocado entre o tubo e a carcaça de fonte de computador.

O cooler foi alimentado por uma fonte 12V ligadas a uma ponte H, controlada por um Arduino, alimentado diretamente pelo computador, que também recebe a posição de um esfera colocada dentro do tubo por meio de um sensor ultrassônico.

Os materiais utilizados estão listados abaixo.



(a) Arduino



(b) Ponte H



(c) Sensor ultrassônico



(d) Cooler 12V

Os dados coletados por meio da IDE do Arduino então foram enviados ao MatLab que utilizou os mesmos para realizar a modelagem do sistema e para o Simulink que projeta e simula os aspectos do sistema e do controlador de acordo com o projeto.

## 3 Metodologia

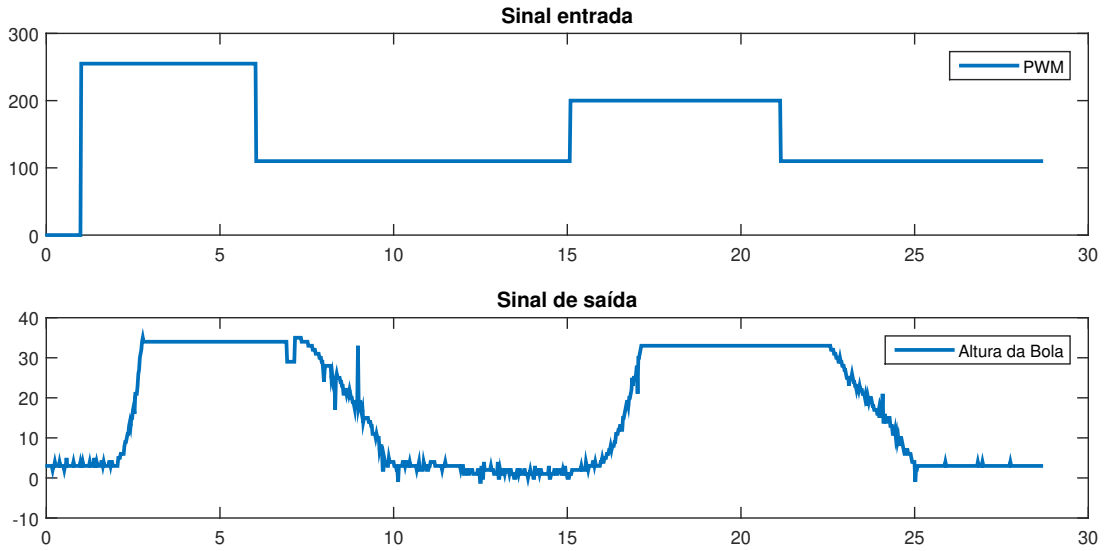
### 3.1 Aquisição e tratamento de dados

Os dados foram coletados a partir de aplicações de degraus em diferentes níveis em cima do sinal de entrada e analisando a posição da bola no tubo. O sinal de entrada variou de acordo com o PWM em 255 níveis onde cada nível corresponde a uma determinada tensão que eleva a bola a uma altura proporcional, com a tensão aplicada no cooler variando de 0V a 12V.

O intervalo de operação do sistema se inicia em um sinal de PWM de 150 e vai até 255. A figura 1 demonstra o ensaio, pode-se observar que a bola atinge a altura máxima com um PWM de 200, sendo que, ao se aplicar um valor de 150, é o suficiente para elevar a bola até o topo. Porém foi mantido o valor máximo do PWM de 255 para o sistema apresentar uma resposta mais rápida. Os dados gerados pela IDE do Arduino foram transformados em um arquivo "dados2.M" com os valores separados em 3 colunas com valores de entradas (PWM), saídas (Altura) e do tempo de amostragem, como demonstrado no código a seguir. Os valores dos arquivos foram então tratados no Matlab para se gerar uma função de transferência do sistema, como mostrado abaixo.

```
1 clc
2 clear all
3 close all
4
5 load dados2.M
6 id = dados2;
7
8 in = id(:,1); % valores de entrada
9 out = id(:,2); % valores de saída
10 t = id(:,3)/1000; % tempo em segundos (Ts = 0.02s)
```

A figura abaixo mostra o ensaio do sistema com a resposta temporal aos degraus aplicados ao PWM na entrada de acordo com os dados do arquivo dados2.m.



**Figura 1:** *Relação temporal dos sinais de entrada e saída do sistema.*

## 3.2 Modelos

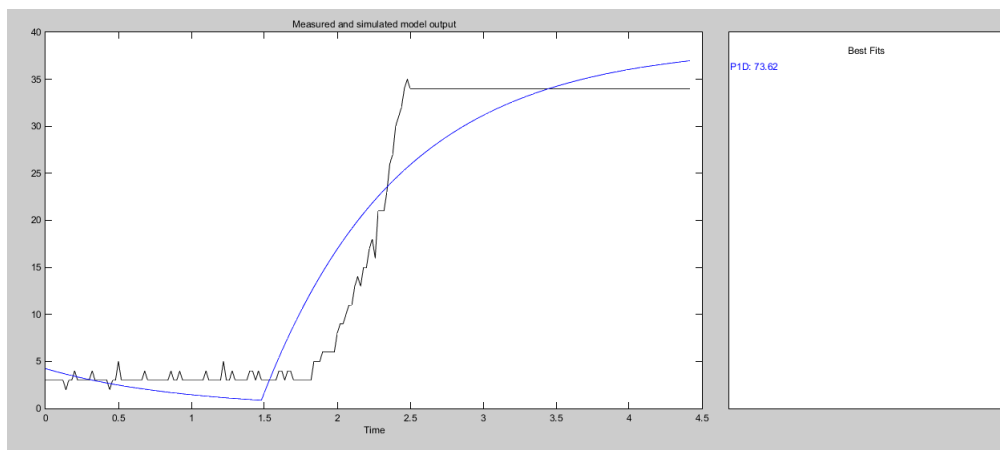
Os dados supracitado foram enviados ao Matlab para então se obter a função de transferência do sistema por meio da ferramenta Toolbox Ident.

A partir da resposta temporal apresentada na figura 1 foram selecionados três degraus que melhor representava a planta para a estimativa da modelagem. Para isso utilizou-se da funcionalidade “Process models” no ident.

Cada um dos degraus selecionados possibilitou estimar uma função de transferência (FT), todas com os mesmos parâmetros: um polo real, um ganho e um delay. Abaixo podemos ver as FTs estimadas, com suas respectivas porcentagem de aproximação em relação ao modelo.

- Primeiro degrau (0-255)

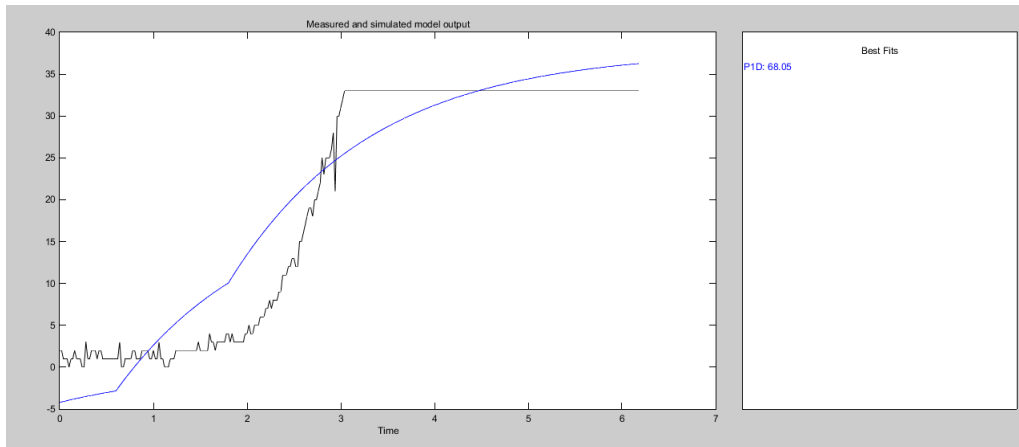
$$T(s) = \frac{0.14103 \cdot e^{-0.6}}{1 + 0.81241s}$$



**Figura 2:** *Relação temporal dos sinais de entrada e saída do sistema.*

- Segundo degrau (110-200)

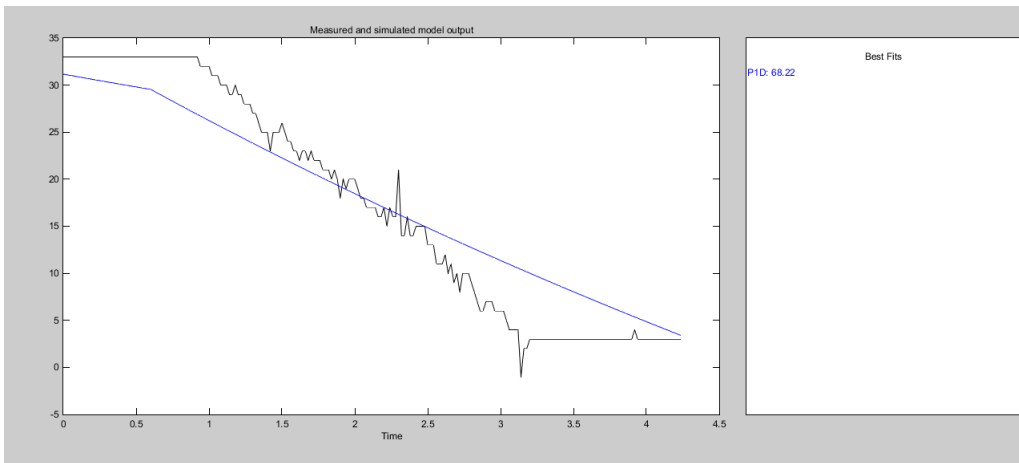
$$T(s) = \frac{0.2131 \cdot e^{-0.6}}{1 + 1.8533s}$$



**Figura 3:** *Relação temporal dos sinais de entrada e saída do sistema.*

- Terceiro degrau (200-110)

$$T(s) = \frac{-0.58967 \cdot e^{-0.6}}{1 + 11.2185s}$$



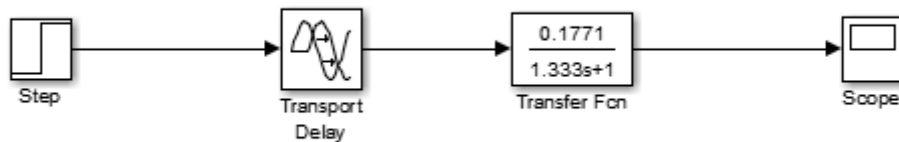
**Figura 4:** *Relação temporal dos sinais de entrada e saída do sistema.*

Foi encontrada a função de transferência, levando em consideração apenas as duas primeiras funções de transferencia, assim apresentando um resultado satisfatório para o sistema. Mesmo apresentando uma resposta conturbada em alguns momentos devido as limitações da robustez do sistema físico, que pode ter influenciado em leituras com pequenos erros agregados a elas, assim gerando alguns erros na saída. Porem o resultado foi satisfatório e a função de transferência encontrada pode ser vista a baixo.

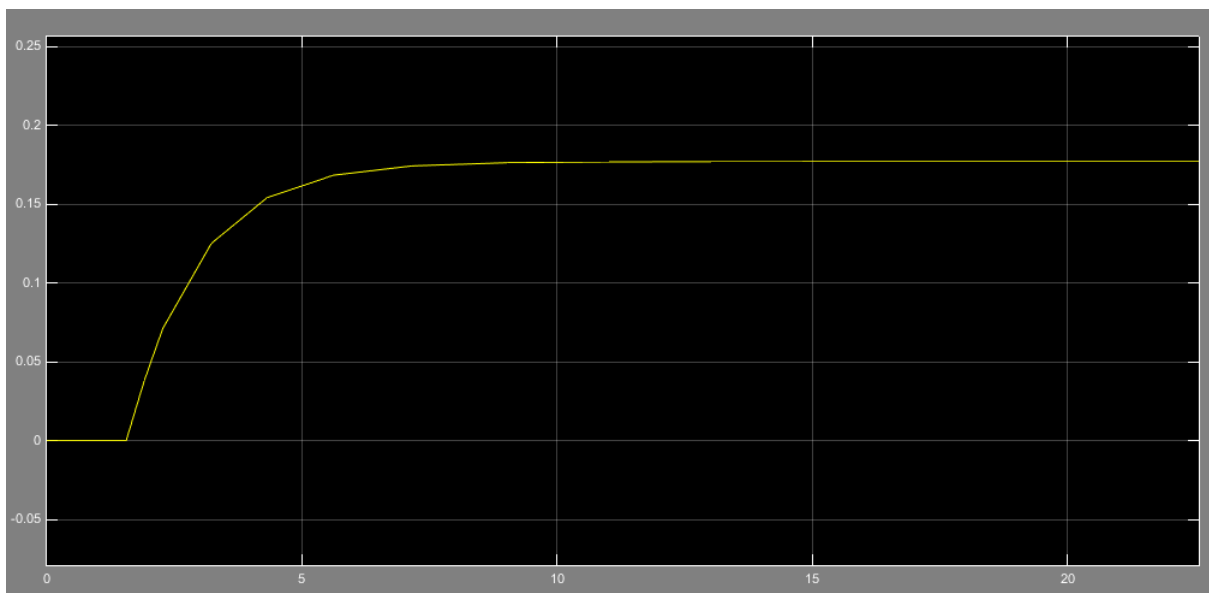
$$T(s) = \frac{0.1771 \cdot e^{-0.6}}{1 + 1.333s}$$

## 4 Controle do sistema

A partir da função de transferência obtida, foi possível projetar um controlador por meio do bloco PID do Simulink. A primeira fase para o projeto do controlador foi o esboço da malha aberta do sistema, conforme a figura 5, com objetivo de observar a resposta do sistema ao degrau unitário, mostrada na figura



**Figura 5:** *Design do sistema em malha aberta.*

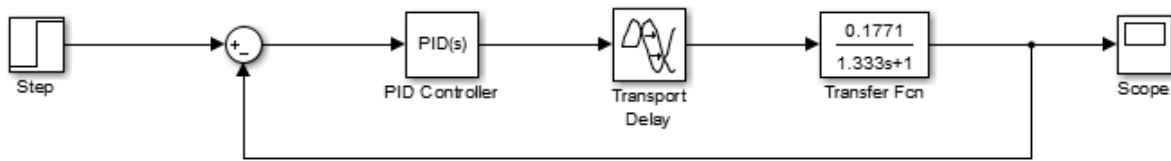


**Figura 6:** *Resposta do sistema em malha aberta ao degrau unitário.*

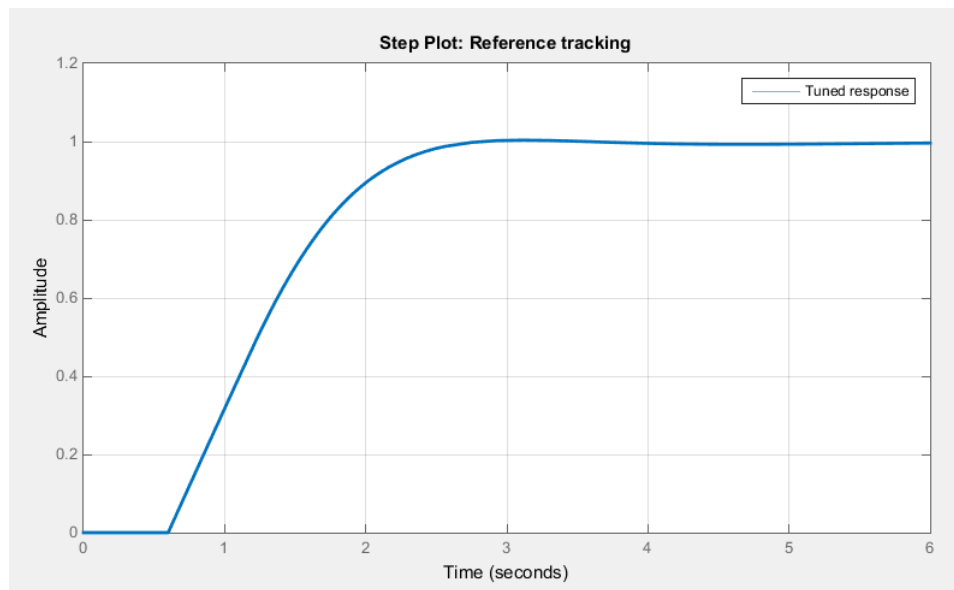
Como pode ser observado na figura 6 o tempo de acomodação é muito longo devido a falta da realimentação do sistema. A partir deste gráfico foram determinados os parâmetros desejados de Overshoot e tempo de acomodação abaixo de 10% e 5 segundos respectivamente, ainda apresentando um erro em regime permanente próximo a zero.

### 4.1 Parametrização do PID

Utilizando o PIDTuner disponível no MatLab foi possível fazer o controle do sistema, alterando o tempo de resposta do sistema e o comportamento transitório, assim projetando um controlador que apresenta um overshoot e tempo de acomodação que estão dentro dos padrões desejados.



**Figura 7:** Design do sistema em malha fechada.



**Figura 8:** Resposta ao degrau PID.

Controller Parameters	
	Tuned
Kp	6.0154
Ki	4.2507
Kd	
Tf	
Performance and Robustness	
	Tuned
Rise time	1.29 seconds
Settling time	2.47 seconds
Overshoot	0.36 %
Peak	1
Gain margin	10.4 dB @ 2.64 rad/s
Phase margin	65 deg @ 0.777 rad/s
Closed-loop stability	Stable

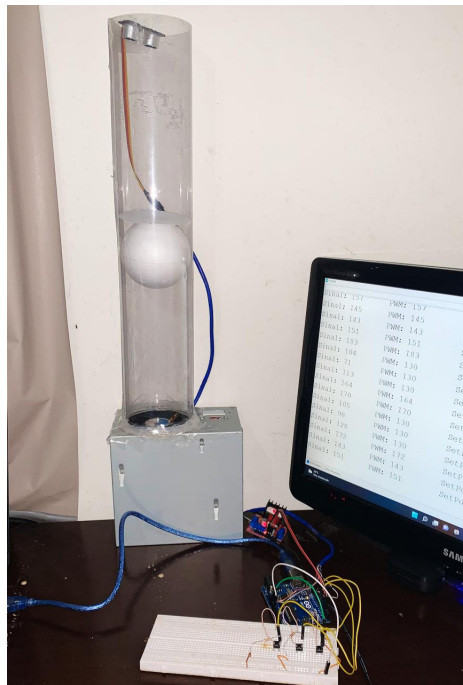
**Figura 9:** Desempenho do controlador.

Como pode-se observar pela imagem 9, nota-se que os requisitos de desempenho foram atendidos, com um overshoot de 0.36%, um tempo de subida de 2.47 segundos e um erro em

regime permanente nulo, apesar de elevar o valor do overshoot. Assim como na modelagem do sistema, o controle do sistema foi satisfatório levando em consideração as limitações do projeto físico já citados à cima.

## 4.2 Controle da planta física

A IDE do arduino foi utilizada para controlar a altura da bola no sistema por meio de um controlador PI. Utilizando a classe PID foi possível controlar o sistema que recebe uma altura desejada (setpoint) e a mantém na mesma até que um dos tres push buttons for pressionado. Para isso foi utilizada a biblioteca NewPing.h para realizar as leituras dos sensores ultrassônicos, como valor de feedback da malha fechada. Os push buttons ligados ao arduino são responsáveis por setar uma nova altura para a bola, cada um com um valores diferente (15 cm, 20cm e 30cm).



**Figura 10:** *Protótipo físico do sistema.*

## 5 Resultados e discussões

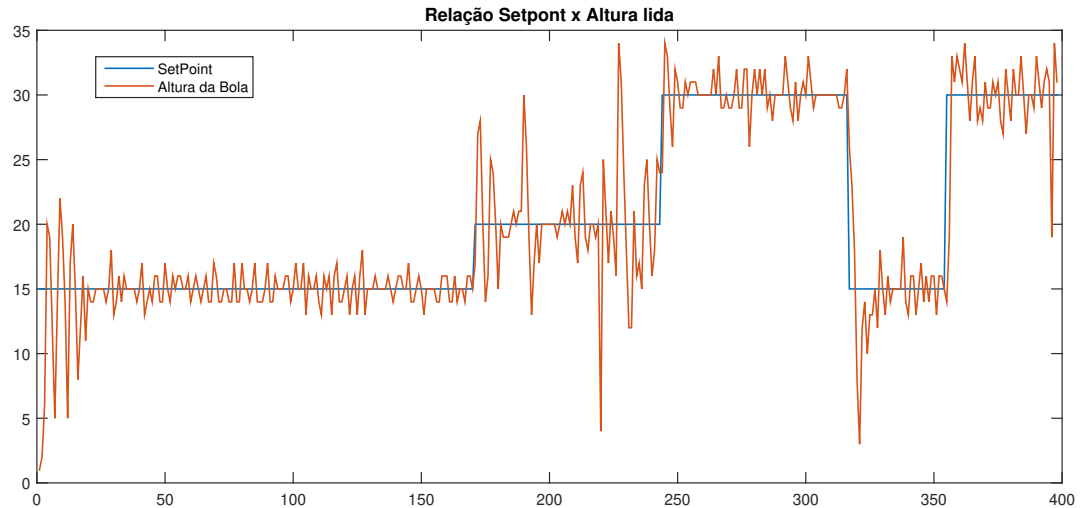
O experimento se propunha a controlar a altura de uma bola em tubo por meio de um cooler que suspende a mesma, e os resultados esperados foram satisfatórios. Assim como proposto o sistema estabilizou a bola em uma determinada posição de forma consistente para qualquer entrada e alcançou os parâmetros de overshoot, tempo de acomodação e erro permanente desejados.

As figuras 11 e 12 abaixo demonstram o desempenho do protótipo, e podemos ver que a planta física tem muito mais variações do que no sistema simulado, assim gerando um erro maior do que o descrito acima. Tais diferenças ocorrem devido a natureza física do sistema que esta

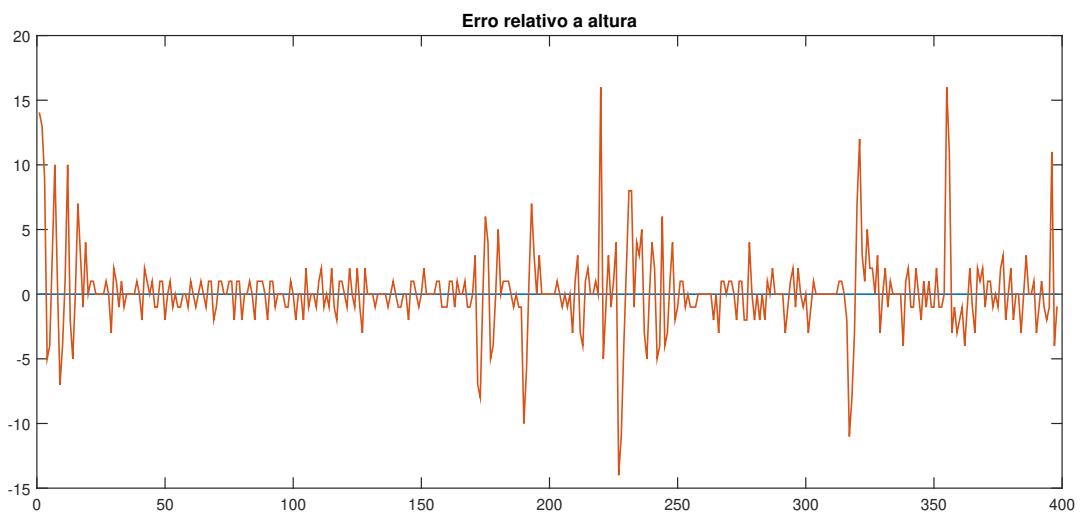


sujeito a inúmeras perturbações as quais não foram acontecidas ao se fazer a modelagem do sistema.

Entretanto os resultados foram satisfatórios com a bola se estabilizando com apenas um pequeno erro de mais ou menos 2 cm.



**Figura 11:** *Desempenho do protótipo .*



**Figura 12:** *Erro*

## 6 Conclusão

O projeto não só foi uma ótima oportunidade de aplicar os conceitos de controle e modelagem em uma abordagem prática, mas também aplicar conhecimentos de diferentes outras áreas da engenharia aprendidas na universidade.

Apesar de ter uma resposta satisfatória, ainda existem melhorias que poderiam ser aplicadas ao trabalho, como a utilização de um sensor mais preciso para esta aplicação, como um sensor

infravermelho, uma vez que utilizar o sensor ultrassônico em um ambiente com vibrações e dentro de um recipiente pequeno acabou gerando alguns erros que não foram possíveis de serem anulados no resultado final. Outro causador de problemas para o projeto foi a construção do protótipo que apresentava falhas inerentes à forma com que foi construído, de forma manual e sem a utilização de ferramental de maior precisão, assim dificultando a obtenção de dados mais precisos.

Apesar das dificuldades já descritas, com base nas simulações realizadas e nos testes físicos, podemos dizer que o controlador funciona adequadamente, e independentemente das perturbações, ele conseguiu controlar adequadamente o sistema.

## 7 Referências

- LOGATA, Katsuhiko. Engenharia de controle moderno. 5. ed. São Paulo,SP:Pearson,2011
- Sistemas Não Lineares Sistema Bola no Tubo com Feedback Linearization Disponível em:  
<https://www.youtube.com/watch?v=iJZH6cDkO0Mab>*channel = HawkLab*  
Acesso em: 20 de abril de 2022.
- Introduction to System Identification Disponível em:  
<https://www.mathworks.com/videos/introduction-to-system-identification-82424.html>  
Acesso em: 23 de abril de 2022.