

# Controle de Nível

Edison Macedo de Melo Júnior, Leonardo Fransoze Soriani, Murilo Henrique Gomes Ferreira.

**Resumo** — Este artigo aborda o controle de nível em sistemas lineares, o qual mostra em detalhes a planta (mostrando as suas dependências físicas e particularidades), o sensor (determinado o seu tipo e comportamento) e o atuador (detalhando a sua capacidade e comportamento) que foram usados. As funções de transferência para cada componente são determinadas ao longo do artigo e as demais considerações são feitas para os circuitos e algoritmos auxiliares utilizados (principalmente aqueles no Matlab, no Scilab e no Arduino). A compreensão desses elementos é fundamental para o bom desenvolvimento do projeto e para um resultado satisfatório.

**Palavras-chave**— Controle de nível. Sistemas lineares. Função de transferência.

## I. INTRODUÇÃO

A medição e controle precisos do nível de fluidos são essenciais em inúmeras aplicações industriais, desde sistemas de abastecimento de água e tratamento de efluentes até processos de controle de um reservatório em um posto de gasolina. A capacidade de manter um nível desejado com precisão é um elemento crítico para garantir a segurança, a eficiência operacional e a qualidade do produto em uma ampla gama de processos.

Neste artigo, exploraremos o processo de como realizar o controle de nível em sistemas mais simples e lineares, bem como as estratégias e métodos utilizados para alcançar essa tarefa de forma assertiva. Da teoria à prática, este artigo fornecerá uma visão sobre o conteúdo que vimos em sala e do procedimento para se modelar um sistema de controle.

## II. PLANTA

A planta desenvolvida consiste em dois tanques, com material de acrílico, e um deles acoplado na lateral acima do outro, de modo que uma torneira foi colocada ao final do tanque de cima para realizar o esvaziamento do mesmo.

Para realizar o sensoriamento do nível foi utilizado a tensão de saída de um potenciômetro como referência. O potenciômetro foi fixado na parte superior do tanque de cima de forma que sua roldana de ajuste estivesse ligada a uma haste de alumínio presa a um isopor de forma esférica (o que seria a bóia do sistema).

Dessa forma, conforme o nível de água sobe, a bóia gira o potenciômetro através da haste, alterando o nível de tensão que é enviado à entrada analógica do Arduino. Para realizar a aferição do nível em que se encontra o fluido no reservatório foi utilizado uma régua de 20 centímetros (cm) fixada em uma das paredes do tanque.

A seguir é possível ver algumas imagens que corroboram com o entendimento de como é a planta (como a régua faz parte dos equipamentos para medições das variáveis do sistema, sua imagem encontra-se no item III - A).



Figura 1. Torneira para esvaziamento do tanque.

Abaixo há uma representação geral da planta, com todos seus componentes.



Figura 2. Planta do sistema.

A tabela abaixo mostra a relação dos materiais utilizados na construção da planta.

Tabela 1. Materiais utilizados na planta.

Materiais Utilizados
1 torneira.
1 régua com extensão de 20 centímetros.
2 tanques de acrílico com aproximadamente 5 litros (L) de capacidade.

### III. MODELAGEM DOS COMPONENTES DO SISTEMA

#### A. Equipamento para medição e monitoramento das variáveis do sistema

Os equipamentos utilizados para as medições foram o Arduino, para receber os valores de tensão pela entrada analógica, e uma régua, utilizada para a verificação física do nível do tanque.

Na figura abaixo temos a régua que foi utilizada para verificar o nível em que se encontra o fluido no tanque.



Figura 3. Régua para aferição do nível.

Abaixo há a representação do Arduino que foi usado para receber os valores na simulação realizada.

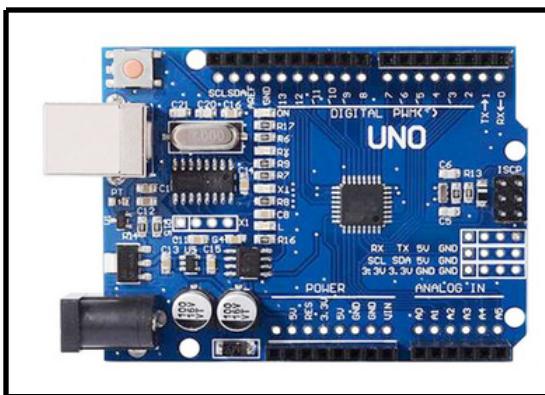


Figura 4. Arduino Uno usado para registros dos testes práticos.

Abaixo há a relação dos códigos utilizados na interface de desenvolvimento do Arduino, no *Matlab* e no *Scilab*.

Algoritmo implementado no ambiente de desenvolvimento integrado (IDE, *Integrated Development Environment*) do Arduino:

Algoritmo 1: leitura do conversor analógico digital do arduino a cada 100ms.

```
// Pino analógico conectado ao potenciômetro.
const int potPin = A5;
void setup() {
  Serial.begin(9600); // Inicia a comunicação serial.
}

void loop() {

// Lê o valor do potenciômetro.
int sensorValue = analogRead(potPin);
```

// Converte o valor para tensão (0-5V).

float voltage = sensorValue \* (5.0 / 1023.0);

// Envia a tensão lida para o MATLAB.

Serial.println(voltage);

// Aguarda um curto período de tempo (em milissegundos) para evitar sobrecarga na comunicação serial.

delay(100);

}

Para melhor ilustrar como foi feita as ligações do Arduino com o potenciômetro, para leitura dos dados nos ensaios da função de transferência do tanque, foi realizado um desenho esquemático no software *TinkerCAD*. O mesmo encontra-se abaixo. A saída do componente resistivo foi ligada na entrada analógica (A5) do Arduino para receber os dados; Os dois outros pinos do componente foram ligados na alimentação (*V<sub>in</sub>*, com valor nominal de 5 V) e no GND (*ground*, “terra”).

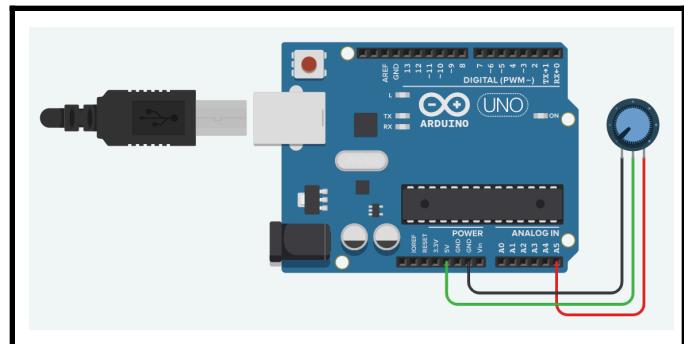


Figura 5. Esquemático das ligações feitas no Arduino junto com o potenciômetro.

Algoritmo 2: leitura da porta serial do arduino pelo *Matlab* e plotagem em tempo real.

```
% O comando abaixo irá encerrar e excluir qualquer
% objeto de comunicação serial associado à porta 'COM7'.
delete(instrfind({'Port'},{'COM7'}));
close % Fecha todas as janelas de figuras abertas.
clear % Limpa as variáveis da memória.
clc % Limpa comandos/dados anteriores no prompt
% sem apagar da memória.
```

% Configuração da porta serial.

serObj = serialport("COM7", 9600);

% Define o caractere de terminação.

configureTerminator(serObj, "CR/LF");

% Configuração da plotagem do gráfico.

figure;

xlabel('Tempo (s)');

ylabel('Tensão');

title('Tensão ao Longo do Tempo');

```
% Inicialização dos vetores de tempo e dados.
time = [];
data = [];

% Tempo total de coleta de dados.
tempo_total = 1800; % tempo em segundos (equivalente a 30 minutos).

% Loop para receber dados e criar o gráfico.
tic; % Inicia a contagem do tempo.
while tic < tempo_total
    sensorValue = str2double(readline(serObj));
    time(end + 1) = tic; % Tempo decorrido em segundos.
    data(end + 1) = sensorValue;
    plot(time, data, '-o'); % Atualiza o gráfico em tempo real.
    ax = gca;
    % Exibe 6 casas decimais após a vírgula.
    ax.YAxis.Exponent = -6;

    % Atualiza o gráfico em tempo real.
    xlabel('Tempo (s)');
    ylabel('Tensão (V)'); % Unidades de medida dos eixos.
    title('Tensão ao Longo do Tempo');
    drawnow; % Atualiza o gráfico em tempo real.
end

clear serObj; % Fecha a conexão da porta serial.
```

Algoritmo 3: simulação teórica da planta feita no Scilab.

```
s = poly(0, "s"); // Definição da variável complexa "s".

// Definição das constantes/funções a serem usadas no sistema.
A = 6;
Km = 0.8605;
Kse = 0.3634;
G = (0.00008688)/(s+0.004075);
FT = syslin('c', (Km*Kse*G));

// Vetor de tempo, início em 0 passos de 0,01 até 10000 segundos.
t = 0:0.01:10000;
// Saída do sistema com uma entrada em degrau
R = csim('step', t, FT);
plot(t, A*R) // Plotagem do gráfico da resposta.
xgrid;
```

Algoritmo 4: plotagem dos resultados práticos e teóricos para fazer uma comparação.

```
% Importa a figura correspondente ao ensaio de malha aberta.
fig = openfig("planta grafico.fig");
axObjs = fig.Children;
dataObjs = axObjs.Children;

% define os valores da figura nos eixos para a plotagem
```

```
x = dataObjs(1).XData;
y = dataObjs(1).YData;
x = x-122;
figure(2)
plot(x, smooth(y))
hold on

% define os coeficientes do sistema do tipo [s^2 s^1 s^0]/[s^2 s^1 s^0].
sys1 = tf((0.00008688*0.3634*0.8605), [1 0.004075]);

% amplitude do degrau utilizado.
opt = stepDataOptions('StepAmplitude', 6);
step(sys1, opt, 'r')
xlim([0 1500]);
ylim([0 0.055]);
title('Resposta ao Degrau');
xlabel('Time', 'FontSize', 13);
ylabel('Amplitude', 'FontSize', 13);
legend('FT prática', 'FT simulada', 'Location', 'southeast');
hold off
```

#### B. Função de transferência do sensor

O sensor utilizado na planta foi um potenciômetro de 100k $\Omega$ . O potenciômetro é um componente eletrônico que possui sua resistência elétrica ajustável devido a sua roldana. Esse componente possui três terminais e atua como um divisor de tensão quando usado todos eles. Dessa forma, a cada nível do líquido que tínhamos no tanque, um valor de tensão diferente era apresentado na saída do componente.



Figura 6. Potenciômetro utilizado como sensor de nível.

Inicialmente foi realizada a montagem de um sensor utilizando um eixo com uma boia em uma extremidade, e acoplado ao potenciômetro na outra extremidade. Ao tentar retirar a função de transferência do sensor, observou-se que conforme o nível do tanque era alterado, um ângulo correspondente era indicado no sensor em forma de tensão, assim, apresentando um funcionamento não linear.

Devido ao comportamento não linear da montagem do sensor com a bóia, foi realizada uma arguição em sala de aula junto com os colegas e o professor. Dessa forma, conseguimos observar que haveria a possibilidade de linearizar o sensor através da montagem de um mecanismo que tornasse o movimento da bóia totalmente linear conforme o nível do tanque subia. Em grupo, pensamos e montamos um mecanismo constituído de uma haste ligada à esfera de isopor (bóia) que, por sua vez, é ligada a um pino e pode se movimentar livremente. Neste pino é ligado uma segunda haste que é fixada no eixo do potenciômetro. Desta forma, a boia faria um movimento fixo de subir e descer conforme o nível e não um ângulo como na configuração anterior do sensor.

Para a obtenção da função de transferência do sensor foi necessária a realização de algumas medidas em relação à altura em que o nível do tanque se encontrava e a tensão que o potenciômetro apresentava.

Desse modo, nos ensaios de teste que foram feitos, foi definido uma tensão de 12 V nos terminais do motor CC somente para encher o tanque que seria aferido o nível. Após isso, foi colocado determinado nível de água no tanque e observado qual valor tensão tinha no componente resistivo. Os valores de altura do tanque variaram de 0 a 9 e a tensão no sensor chegou a 3,635 V.

Abaixo há a foto do sistema durante a realização do ensaio para a obtenção da função de transferência.

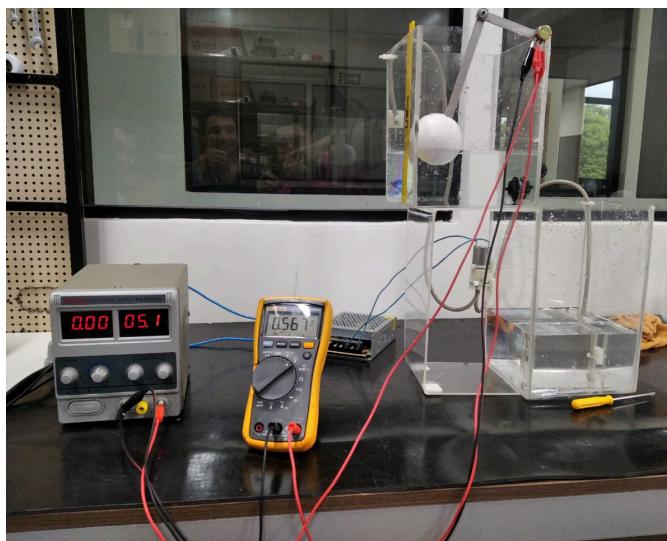


Figura 7. Imagem do sistema durante os ensaios.

Para elucidar os ensaios realizados para a extração da função do sensor segue a tabela e o gráfico correspondentes abaixo, com os valores da altura (em centímetros) e da tensão no sensor (em Volts).

Tabela 2. Ensaios para a Função de Transferência do Sensor.

ENSAIO	ALTURA (cm)	TENSÃO NO SENSOR (V)	FT (V/cm)
1	0	0,001	0
2	1	0,046	0,045
3	2	0,412	0,366
4	3	0,836	0,424
5	4	1,300	0,464
6	5	1,791	0,491
7	6	2,224	0,433
8	7	2,727	0,503
9	8	3,266	0,539
10	9	3,635	0,369
Kse (Constante do Sensor)			0,3634

Abaixo temos o gráfico correspondente à função de transferência do sensor, com os valores do ensaio realizado.

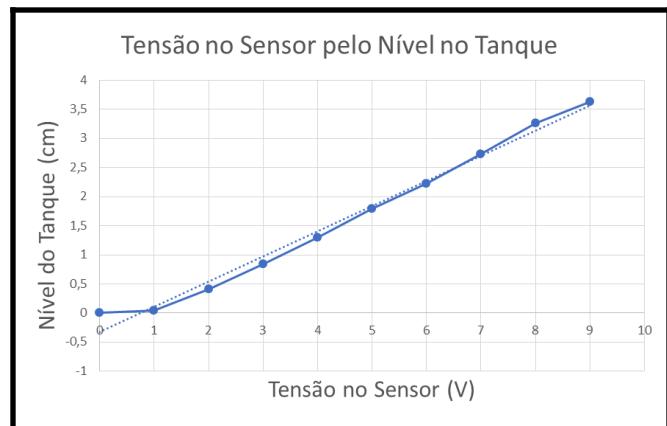


Figura 8. Gráfico referente aos ensaios feitos para a FT do sensor.

Podemos ver uma linha tracejada junto à principal, no gráfico. Essa linha representa a tendência que a tensão do sensor iria seguir pela altura do tanque. Com isso, podemos observar a linearidade do sensor com o seguimento dessa tendência.

Dessa forma, com os valores obtidos pelo ensaio, foi possível obter a constante referente ao sensor (que foi denominada “Kse”), a qual é a média dos valores obtidos fazendo a diferença entre o valor de tensão lido pelo valor de tensão anterior dividido pela diferença entre o nível lido pelo nível anterior. Como temos que a entrada do sensor é o nível do tanque em centímetros e a saída é a tensão em Volts (V),

temos que a unidade dessa função de transferência seria Volts/centímetros.

$$FT_{sensor} = \frac{V_i - V_{i-1}}{N_i - N_{i-1}} = \frac{Volts}{centímetros} = \left[ \frac{V}{cm} \right] \quad [1]$$

$$FT_{sensor} = 0,3634 \left[ \frac{V}{cm} \right]$$

Abaixo podemos ver uma foto do sensor usado (potenciômetro) em sua posição junto a planta. O mesmo foi fixado através de um furo, onde sua roldana está na parte interna do tanque ligada às hastes e seu encapsulamento fica na parte de fora do tanque.

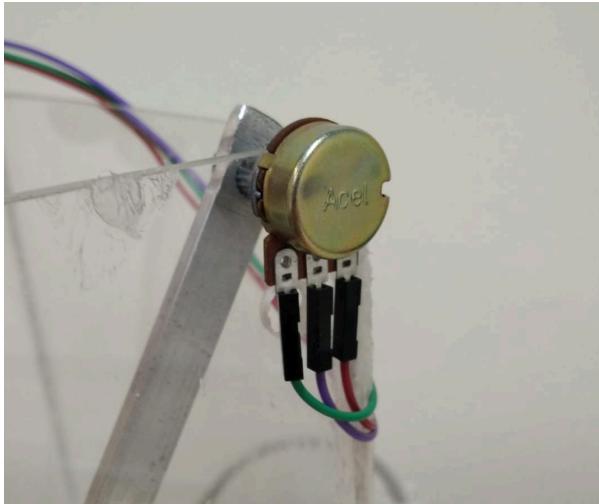


Figura 9. Potenciômetro utilizado como sensor de nível.

Foi feito um furo na placa de acrílico de modo a encaixar a roldana do potenciômetro no orifício e ficar fixo à planta. Abaixo há a imagem da haste e da bóia presas à roldana do sensor.

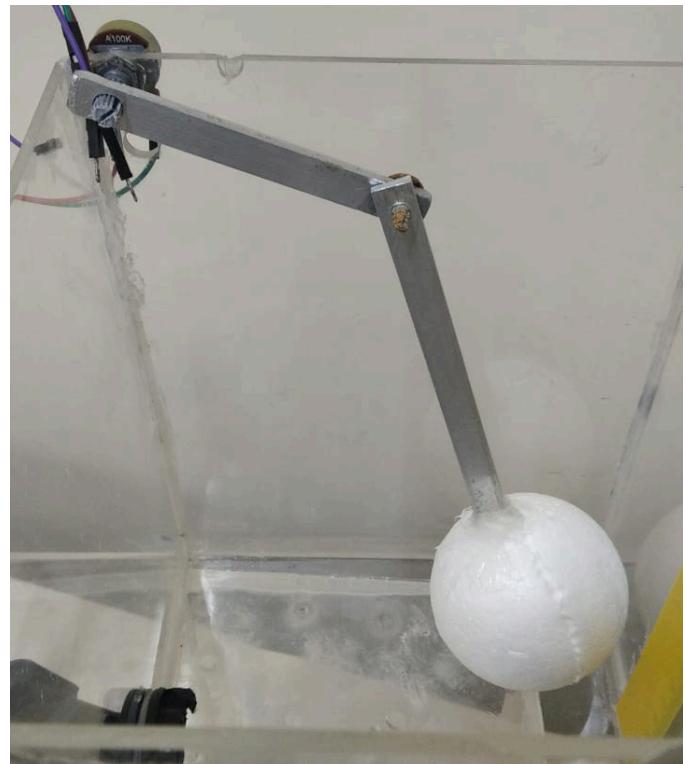


Figura 10. Visão da haste e da bóia ligadas diretamente ao sensor de nível.

#### C. Função de transferência do atuador

Para o nosso controle de nível, foi necessário o uso de uma bomba que transferisse a água do recipiente de baixo para o de cima. Dessa forma, foi escolhida uma bomba de corrente contínua (bomba CC) para ser o atuador do nosso sistema de controle. Essa bomba atua com uma tensão de até 12 V e uma corrente de 6 Ampères. A imagem que se refere ao atuador encontra-se abaixo.

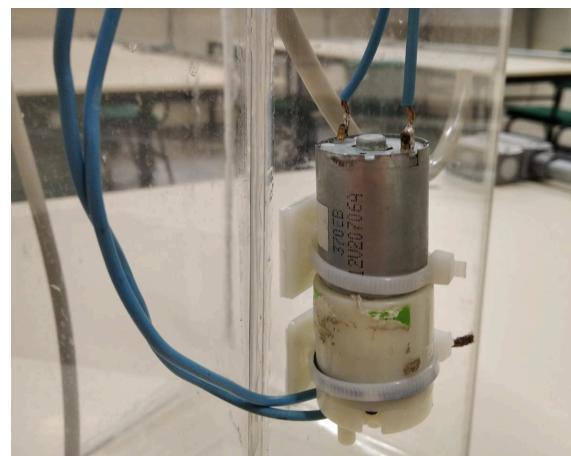


Figura 11. Atuador usado (bomba CC de 12 V/6 A).

Para a obtenção da função de transferência do atuador foi aplicada uma tensão no motor que se manteve constante ao longo do enchimento de 1 litro (L) (o equivalente a 1000 centímetros cúbicos ( $cm^3$ )) de água no tanque superior da

planta. Assim, foi possível observar quanto tempo foi necessário para acrescentar 1 L de água e dividindo os 1000 cm<sup>3</sup> pelo tempo percorrido, podemos encontrar qual foi a vazão que tivemos na saída da bomba. Abaixo podemos ver os ensaios realizados e os resultados obtidos.

$$Vazão = \frac{1000}{t} \quad [2]$$

Tabela 3. Ensaios para a Função de Transferência do Motor.

Função de Transferência do Motor (Km)				
Ensaio	Tensão no Motor $V_m$ (V)	Tempo (s)	Vazão (cm <sup>3</sup> /s)	FT (Vazão pela Tensão)
1	3,00	332,32	3,0091297	1,003043233
2	4,00	255,55	3,9131286	0,978282137
3	6,00	180,08	5,5530875	0,925514586
4	8,00	157,10	6,3653724	0,795671547
5	10,00	137,29	7,2838517	0,72838517
6	12,00	130,47	7,6645972	0,638716435
Média de FT (Km - Constante do Motor)			0,860593066	

Com os valores obtidos da saída pela entrada em cada ensaio, foi possível ser feita uma média desses valores e, assim, encontrarmos uma constante para ser relacionada com o motor/atuador do nosso sistema (a qual foi chamada de “Km”). Dessa forma, podemos analisar a tabela com os valores abaixo.

Tabela 4. Aplicação da Constante do Motor (Km) na Tensão injetada no Motor.

Aplicando a constante Km encontrada	
$V_m$ (V)	Vazão (cm <sup>3</sup> /s)
1,00	0,860593066
2,00	1,721186133
3,00	2,581779199
4,00	3,442372266
5,00	4,302965332
6,00	5,163558399
7,00	6,024151465
8,00	6,884744532

9,00	7,745337598
10,00	8,605930664
11,00	9,466523731
12,00	10,327116797

Abaixo encontra-se o gráfico traçado partindo da tabela da função de transferência do motor.

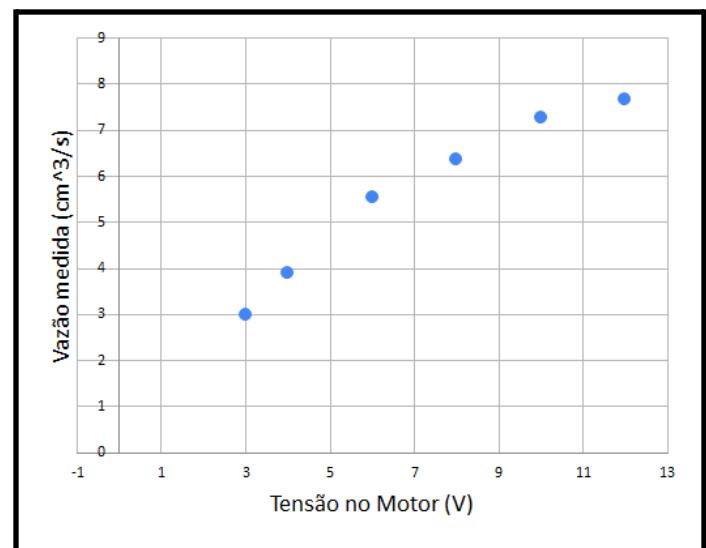


Figura 12. Gráfico da tensão no motor pela vazão medida.

Abaixo encontra-se o gráfico traçado partindo da tabela da constante aplicada ao motor.

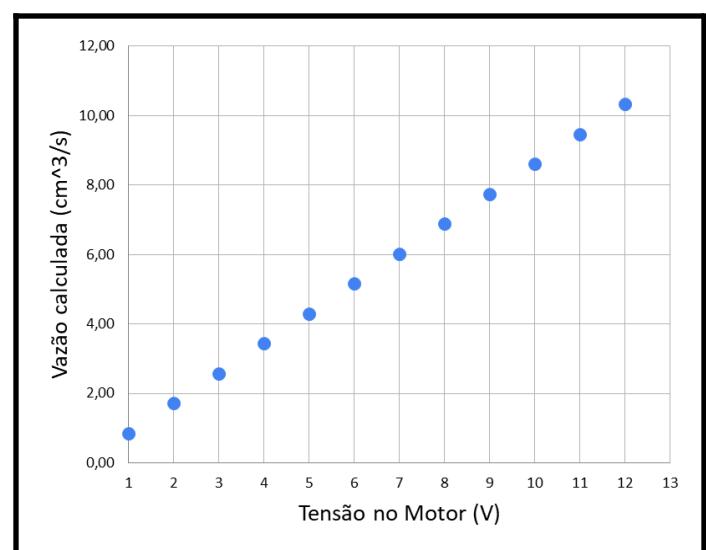


Figura 13. Gráfico da tensão no motor pela vazão calculada.

Determinando a função de transferência do motor, temos:

$$FT_{motor} = \frac{Vazão}{Tensão no Motor} = \left[ \frac{cm^3/s}{V_m} \right] \quad [3]$$

$$FT_{motor} = 0,860593066 \left[ \frac{cm^3/s}{Vm} \right]$$

#### D. Função de Transferência da Planta (Tanque)

Para a extração da função de transferência da planta foi necessária a implementação de um código no *Matlab* em paralelo com a aplicação do *Arduino* para receber os dados que estavam sendo transmitidos para ele. Sendo assim, foi possível a plotagem de um gráfico em concordância com um sistema de primeira ordem estudado na disciplina anteriormente.

$$FT_{sistema} = \frac{A*K}{s+a} = \frac{\text{Nível do Tanque}}{\text{Vazão}} = \left[ \frac{V}{V} \right] [4]$$

Para a plotagem no *Matlab*, foi definido um código que coleta pontos em tempo real a cada 100 milissegundos (ms) em um período de tempo total de 1800 segundos (aproximadamente 30 minutos) e realizado a plotagem em um gráfico. O experimento para a função de transferência da planta ocorreu da seguinte forma: Primeiro uma tensão de 4 V foi aplicada ao atuador e, ajustando a torneira de saída, foi encontrada a vazão que estabiliza o nível do tanque principal no zero da planta definido em 3 cm de altura. Em seguida, um degrau de 6 V foi aplicado ao motor e realizado a coleta em tempo real das mudanças de nível coletadas pelo sensor até a estabilização do tanque em um novo nível.

Vale ressaltar que o experimento foi realizado de tal forma que uma tensão é inserida no atuador e uma tensão é coletada do sensor, portanto, a unidade de medida da função de transferência do sistema é Volt/Volt.

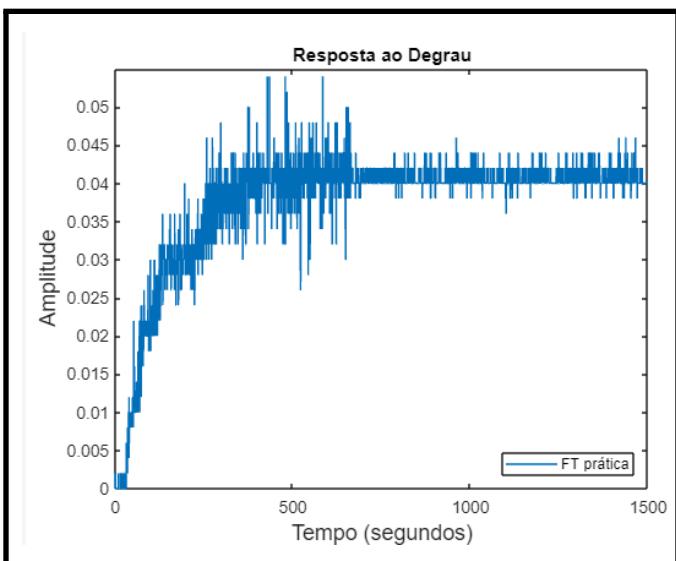


Figura 14. Gráfico extraído pelo *Matlab* mostrando a tensão no sensor ao longo do tempo.

No gráfico, foram obtidos os valores de 0,04 de regime permanente e 245,36 segundos para alcançar 63% desse valor. Para a obtenção da constante “K” da planta foi necessário achar o valor de regime permanente no gráfico e, assim, encontrar o valor que corresponde a 63% disso. O inverso do tempo percorrido para alcançar este valor corresponde a constante “a” da função de transferência. A constante “A” é a diferença do degrau que foi dado no sistema pelo valor de tensão inicial do motor (10 V - 4 V = 6). Com essas duas constantes encontradas foi possível achar uma constante “K<sub>1</sub>” fazendo:  $K_1 = \frac{0,04 \cdot a}{A} = 0,00002717$  que ainda não era o valor final que precisávamos. Após isso, dividimos essa mesma constante pelo produto das constantes do sensor e do motor que foram encontradas (“K<sub>se</sub>” e “K<sub>m</sub>”). Desse modo, foi encontrada nossa constante “K” final da planta com o valor de 0,000086889. Portanto, com o valor de “a” em mãos, a função de transferência do tanque resultou no seguinte equação:

$$FT_{sistema} = \frac{A*K}{s+a} = \frac{0,00002717}{s+0,00407} \left[ \frac{V}{V} \right] [5]$$

$$FT_{planta} = \frac{A*K}{s+a} = \frac{0,000086889}{s+0,00407} \left[ \frac{cm}{cm^3/s} \right] [6]$$

#### IV. RESULTADOS TEÓRICOS DO SISTEMA OPERANDO EM MALHA ABERTA

Abaixo há a representação do sistema em malha aberta. O sistema está na forma de um diagrama de blocos com as funções de transferência do motor, do tanque e do sensor. A implementação foi realizada no *software Simulink*.

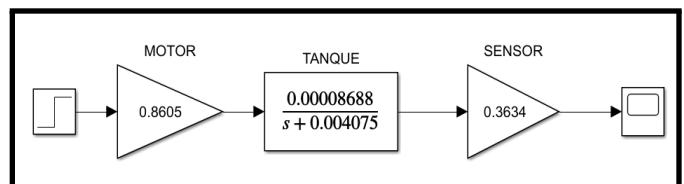


Figura 15. Diagrama de blocos feito no *Matlab - Simulink Academic Use*.

Foi feita a simulação do sistema através do código no *software Scilab* e observado a função de primeira ordem que era o esperado para esse tipo de sistema. Abaixo podemos ver o resultado dessa simulação.

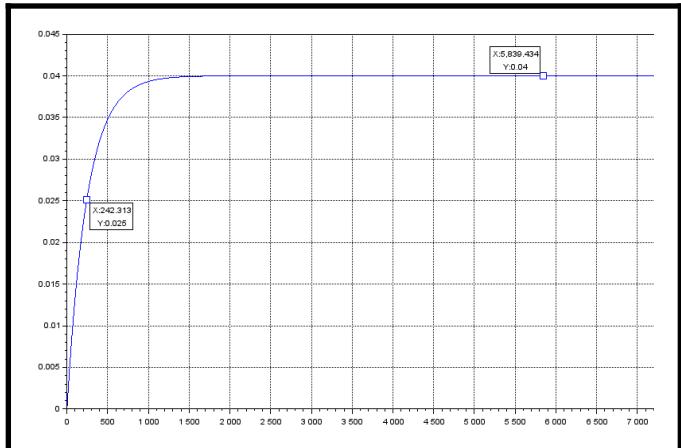


Figura 16. Simulação do sistema de primeira ordem.

Foi observada uma constante de tempo de 242,313 segundos e fazendo o seu inverso, temos 0,004127. O valor final da mostra, em “y”, resultou em 0,04. Foi realizado o mesmo procedimento feito na FT do tanque para achar a constante “K” desse sistema, a qual resultou em um valor de 0,000087982, essa constante se aproxima bastante do resultado prático encontrado.

Abaixo podemos ver a discriminação de todas as funções de transferência do sistema separadas em quatro imagens para melhor visualização.

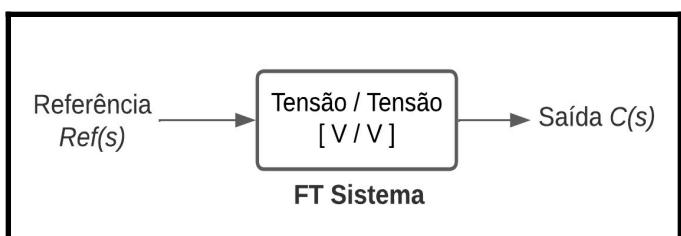


Figura 17(a). Função de transferência de todo o sistema.

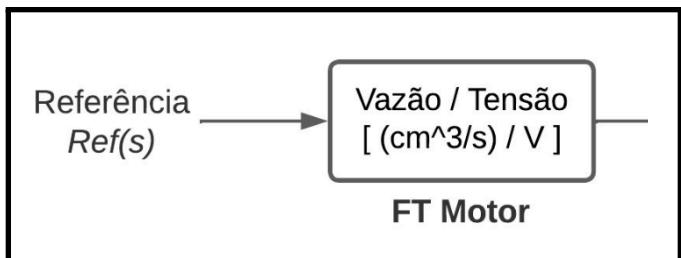


Figura 17(b). Parcela do diagrama correspondente a função de transferência do motor.

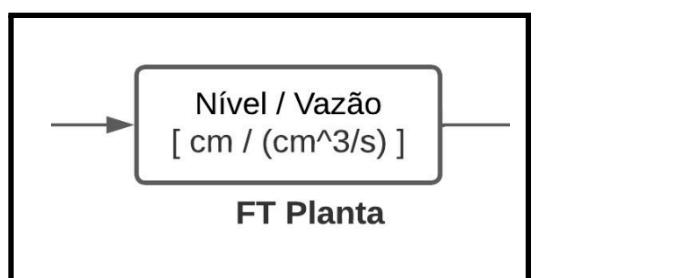


Figura 17(c). Parcela do diagrama correspondente a função de transferência da planta.

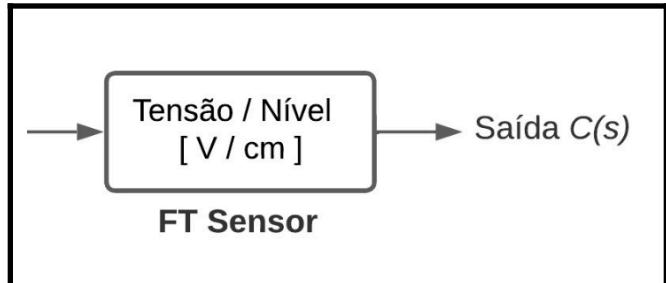


Figura 17(d). Parcela do diagrama correspondente a função de transferência do sensor.

## V. RESULTADOS PRÁTICOS DO SISTEMA OPERANDO EM MALHA ABERTA

O texto contendo o algoritmo desenvolvido para poder ser feito o comparativo entre o ensaio prático realizado no sistema e a simulação em *software* está descrito na seção “III. Modelagem dos Componentes do Sistema”, está denominado como “algoritmo 4”.

Com o auxílio do *software*, conseguimos comparar a curva obtida através do experimento prático com a resposta simulada. Como obtivemos uma simulação em que os valores apresentavam vários picos espúrios, era necessário o tratamento do sinal para conseguirmos uma curva mais condizente com o simulado. Para isso, foi utilizada a função do *Matlab* “smooth”, que suaviza os dados de resposta no vetor usando um filtro de média móvel. O gráfico correspondente a essa implementação encontra-se abaixo.

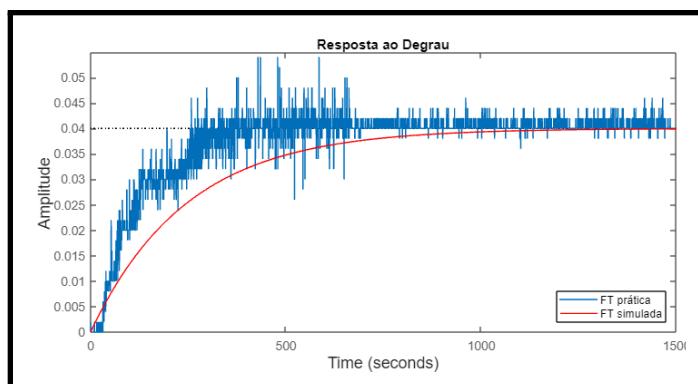


Figura 18. Comparativo dos resultados práticos (obtidos pelo ensaio do sistema) e teóricos (obtido pela simulação no *software*).

Após mais algumas tentativas em tratar a curva obtida pelo ensaio da planta, foi utilizada uma outra função no *software* em questão que retornou um resultado melhor do que o obtido no gráfico anterior. A função usada foi a “smoothdata2” que tem como parâmetros de entrada o gráfico em duas dimensões e o método de suavização que se deseja utilizar.

A mudança que houve no algoritmo para essa implementação foi somente na linha que diz respeito à plotagem do gráfico. Onde tínhamos “plot(x, smooth(y))” passou a ser “plot(x, smoothdata2(y, “sgolay”))”. Após isso, foi obtido o gráfico abaixo, o qual representa o ensaio feito em comparação com a simulação em *software*.

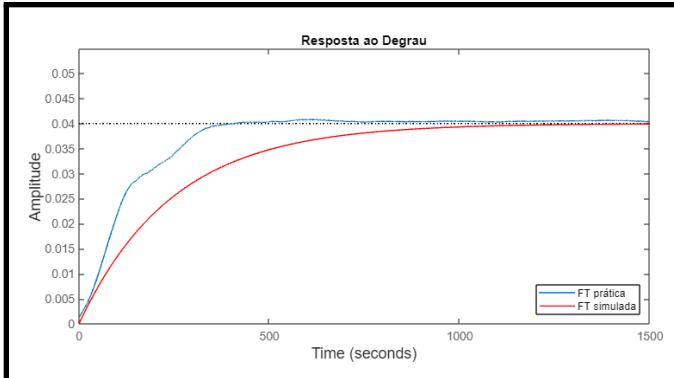


Figura 19. Comparativo dos resultados práticos (obtidos pelo ensaio do sistema) e teóricos (obtido pela simulação no software), com a suavização da curva do ensaio prático.

O valor de regime foi encontrado em 0,04 em uma janela de tempo de 1500 segundos (25 minutos).

Dessa forma, podemos observar que o modelo matemático encontrado é uma boa aproximação do sistema de primeira ordem simulado.

## VI. RESULTADOS TEÓRICOS DO SISTEMA OPERANDO EM MALHA FECHADA

### A. Sintonia do controlador

O controlador escolhido para ser utilizado no projeto foi um proporcional-integral (controlador PI). O controlador em questão combina as ações proporcional e integral no sistema de controle. Isso faz com que a componente proporcional contribua com uma resposta rápida (altera a parte transitória do sistema como sobressinal e tempo de estabelecimento), enquanto a componente integral elimina o erro em regime estacionário, o que melhora a estabilidade e precisão do sistema.

Abaixo há a representação do sistema em malha fechada no programa “Xcos”.

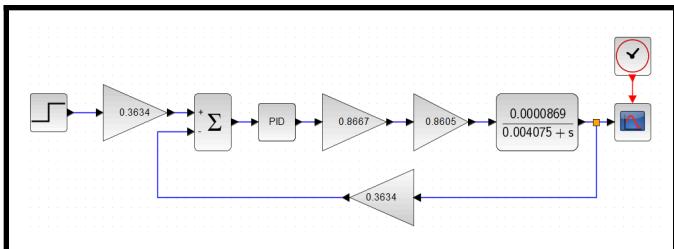


Figura 20. Diagrama de blocos feito no Scilab - Xcos.

Conforme pode ser notado na imagem acima, foi inserido o sensor na entrada logo após a referência em degrau, isso se dá devido ao funcionamento do sistema físico. A referência no circuito físico é um potenciômetro de mesmo valor ao do sensor que a sua tensão é inserida no circuito subtrator, ou seja, usamos a característica do sensor de transformar nível em tensão com referência de entrada para o sistema.

Na simulação, foi inserido o sensor na entrada para, de fato, simular esse comportamento. Dessa forma, foi colocado o

nível como referência através do degrau que é convertido em tensão pelo sensor e gera-se o erro subtraindo essa tensão pela tensão indicada pelo sensor.

Simulando esse diagrama foi possível obter a resposta com o controlador no sistema em malha fechada. Diante disso, podemos ver abaixo a representação em um gráfico do sistema operando em malha fechada.

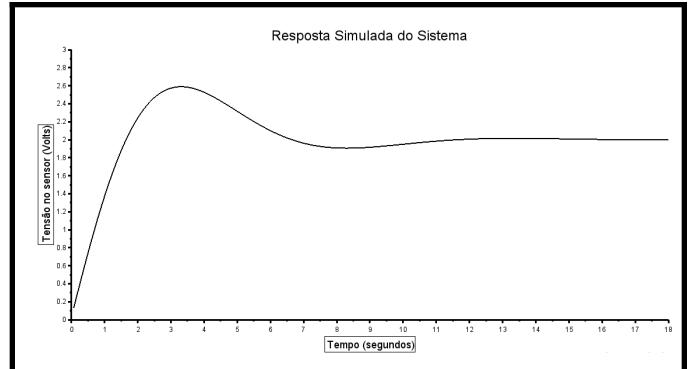


Figura 21. Resposta simulada, representada graficamente, do sistema em malha fechada.

### B. Projeto do circuito analógico do controlador

O primeiro projeto foi a montagem de um circuito gerador de erro (o qual continha tanto componentes passivos como ativos), ou seja, um circuito que realiza a diferença entre a tensão de referência e a tensão do sensor. Esse circuito se dá utilizando uma característica dos amplificadores operacionais, que é fazer a diferença entre a porta não inversora e a porta inversora do dispositivo e resistores para um bom funcionamento do circuito. Toda a simulação dos circuitos físicos foi realizada no software gratuito de simulação LTSpice.

Abaixo temos a representação do circuito gerador de erro.

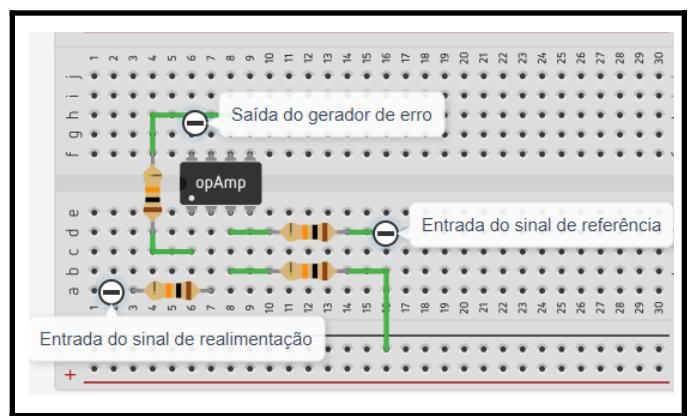


Figura 22. Circuito gerador de erro feito no software TinkerCAD.

A alimentação simétrica do “ampOp” foi omitida a fim de não poluir muito a imagem em questão. Para alimentar o CI é necessária as alimentações nos pinos 7 (alimentação positiva da fonte, “+Vcc”) e 4 (alimentação negativa da fonte, “-Vcc”).

Tabela 5. Componentes utilizados para o gerador de erro.

Componentes Utilizados
1 Amplificador operacional LM741.
4 Resistores de 10 kΩ.

Para o controlador, primeiro foi inserida ao diagrama de blocos de malha aberta a equação característica de um controlador proporcional-integral-derivativo (PID) com a constante derivativa nula para obtermos um controlador proporcional-integral (PI), em seguida foi realizada as seguintes simplificações, a fim de encontrar uma expressão que pudesse calcular as constantes  $K_i$  e  $K_p$  do controlador.

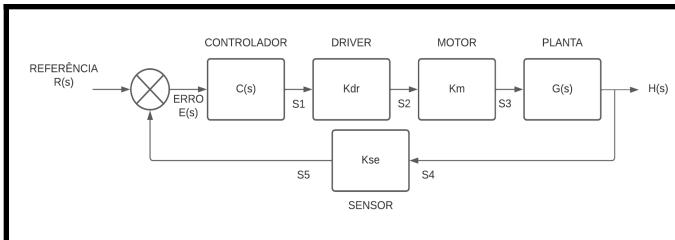


Figura 23. Diagrama de blocos controle em malha fechada.

Abaixo há os equacionamentos que foram necessários para o desenvolvimento:

$$S_4 = E(s) * C(s) * K_{DR} * K_M$$

$$E(s) = R(s) - S_5 = R(s) - H(s) * K_{se}$$

Sinal que entra na planta:

$$S_3 = C(s) * K_{DR} * K_M * (R(s) - H(s) * K_{se})$$

$$H(s) = G(s) * S_3$$

$$H(s) = G(s) * (C(s) * K_{DR} * K_M * (R(s) - H(s) * K_{se}))$$

Saída da planta:

$$H(s) = G(s) * (R(s) * C(s) * K_{DR} * K_M - H(s) * C(s) * K_{DR} * K_M * K_{se})$$

$$H(s) * (1 + G(s) * C(s) * K_{DR} * K_M * K_{se}) \dots$$

$$\dots = R(s) * C(s) * G(s) * K_{DR} * K_M$$

Função de transferência de Malha Fechada:

$$\frac{H(s)}{R(s)} = \frac{G(s) * C(s) * K_{DR} * K_M}{1 + G(s) * K_{DR} * K_M * K_{se}}$$

Equação do controlador:

$$C(s) = (K_p + K_D s + \frac{K_i}{s}) = \frac{K_p s + K_i}{s}$$

$$\frac{H(s)}{R(s)} = \frac{G(s) * \left( \frac{K_p s + K_i}{s} \right) * K_{DR} * K_M}{1 + G(s) * K_{DR} * K_M * K_{se}}$$

$$\frac{H(s)}{R(s)} = \dots$$

$$\dots \frac{86.88\mu * (K_p s + K_i) * K_{DR} * 0.8605}{s * (s + 4,075m) + 86.88\mu * (K_p s + K_i) * K_{DR} * 0.8605 * 0.3634}$$

$$= \frac{86.88\mu * (K_p s + K_i) * K_{DR} * 0.8605}{s + (4,075m + 86.88\mu * K_p * K_{DR} * 0.8605 * 0.3634) * s + 86.88\mu * K_i * K_{DR} * K_M * K_{se}}$$

Um sistema de segunda ordem tem a seguinte característica:

$$\frac{H(s)}{R(s)} = \frac{\alpha \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2}$$

$$2\zeta \omega_n = 0,004075 + 86.88\mu * K_p * K_{DR} * 0.8605 * 0.3634$$

Expressão para a constante proporcional:

$$K_p = \frac{2\zeta \omega_n - 0,004075}{0,00008688 * K_{DR} * 0,8605 * 0,3634}$$

$$\omega_n^2 = 0,00008688 * K_i * K_{DR} * 0,8605 * 0,3634$$

Expressão para a constante integradora:

$$K_i = \frac{\omega_n^2}{0,00008688 * K_{DR} * 0,8605 * 0,3634}$$

Arbitrando valores de projeto:

PO% = 16 (porcentagem de overshoot de dezesseis por cento). Para uma PO% de 16% temos um coeficiente de amortecimento ( $\zeta$ ) igual a:

$$\zeta = -\frac{\ln(\frac{PO\%}{100})}{\sqrt{\pi^2 + \ln(\frac{PO\%}{100})^2}} = 0.5038681$$

$$T_p = 5 \text{ s} \quad (\text{tempo de pico de cinco segundos})$$

$$\omega_n = \frac{\pi}{T_p \sqrt{1 - \zeta^2}} = 0.7274052$$

Substituindo nas expressões, podemos encontrar as constantes do compensador:

$$K_p = 30958.353$$

$$K_i = 22471.312$$

Para a confecção do controlador foram utilizados tanto componentes passivos (resistores e capacitores) como componentes ativos (amplificadores operacionais). Os amplificadores utilizados na montagem foram os circuitos integrados (CI's) LM471 e LM324, assim como diversos resistores e um capacitor. Logo abaixo podemos ver como foi realizada a montagem do controlador na placa de ensaio.

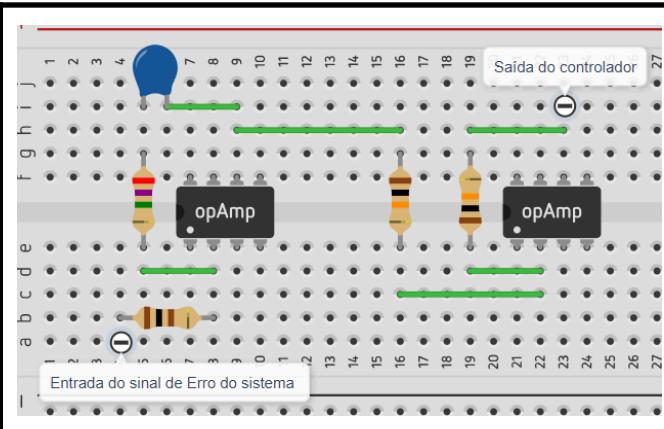


Figura 24. Circuito controlador PI feito no software TinkerCAD.

As alimentações simétricas dos “ampOp’s” foram omitidas a fim de não poluir muito a imagem em questão. Para alimentar os CI's são necessárias as alimentações nos pinos 7 (alimentação positiva da fonte, “+Vcc”) e 4 (alimentação negativa da fonte, “-Vcc”).

Para encontrarmos a relação de resistores e capacitor que foram usados utilizamos as seguintes equações:

$K_p = \frac{R_2}{R_1}$ , para o ganho proporcional. E, para o ganho integrador, temos:  $K_i = \frac{1}{R_1 \cdot C_1}$ , onde “R1” é o resistor que recebe o sinal de erro do sistema.

Dessa forma, para ser definido o valor a ser usado para o “R1”, foi escolhido o valor do capacitor (470 nF) e substituído o valor já encontrado da constante integradora. Isso resultou no valor de 100 Ω para o resistor. Assim, foi possível isolar o “R2” na equação do  $K_i$  e achar seu valor, logo já tínhamos

o valor da constante integradora. Podemos ver toda a relação de componentes e suas particularidades pela tabela abaixo.

Tabela 6. Componentes utilizados para o controlador.

Componentes Utilizados
1 Resistor de 100 Ω.
1 Resistor de 2,7 MΩ.
1 Capacitor de 470 nF.
2 Amplificadores operacionais LM741.
2 Resistores de 10 kΩ.

Para a visualização do circuito prático, abaixo há representação do mesmo montado na placa de prototipação.

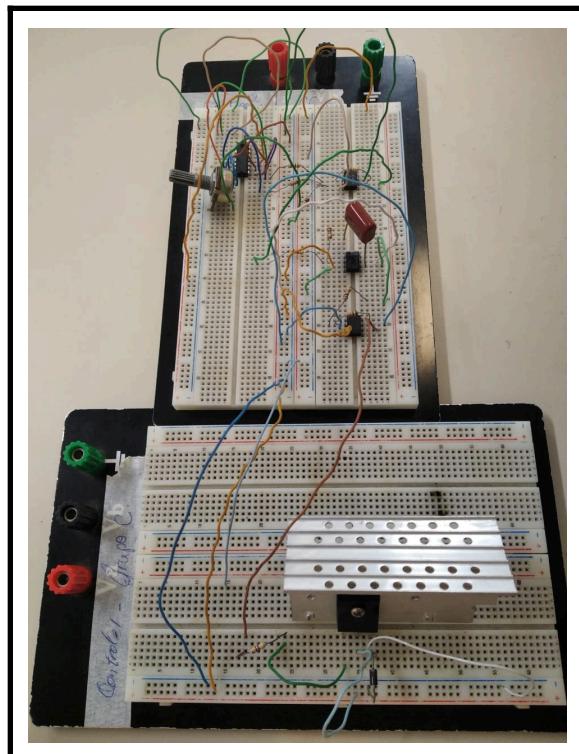


Figura 25. Circuito completo montado na protoboard.

Devido a potência dissipada pelo transistor TIP142, foi inserido um dissipador de calor em seu encapsulamento (o que seria a placa de metal volumosa no canto inferior direito da imagem) para evitar o superaquecimento e possíveis danos ao componente. Esse transistor foi usado para realizar o *driver* de acionamento do atuador (motor) do sistema. A relação dos componentes utilizados no *driver* encontra-se abaixo.

Tabela 7. Componentes utilizados para o *driver*.

Componentes Utilizados
1 Resistor de 100 $\Omega$ .
1 diodo 1N4007.
1 transistor NPN TIP142.

Para uma melhor visualização do circuito de acionamento do motor abaixo há sua representação realizada no *software*.

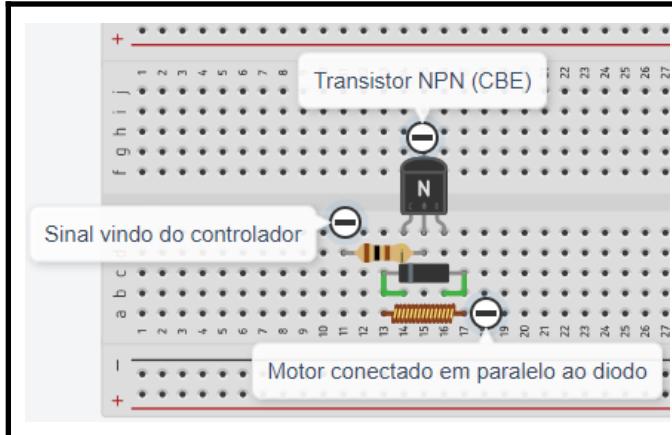


Figura 26. *Driver* para acionamento do motor feito no *software* TinkerCAD.

## VII. RESULTADOS PRÁTICOS DO SISTEMA OPERANDO EM MALHA FECHADA

Para visualizarmos em forma de gráfico o sistema sendo controlado, foi realizado um teste em malha fechada do mesmo e extraídos os dados que estavam sendo observados da tensão de saída no sensor. Com isso, obtivemos a curva correspondente abaixo.

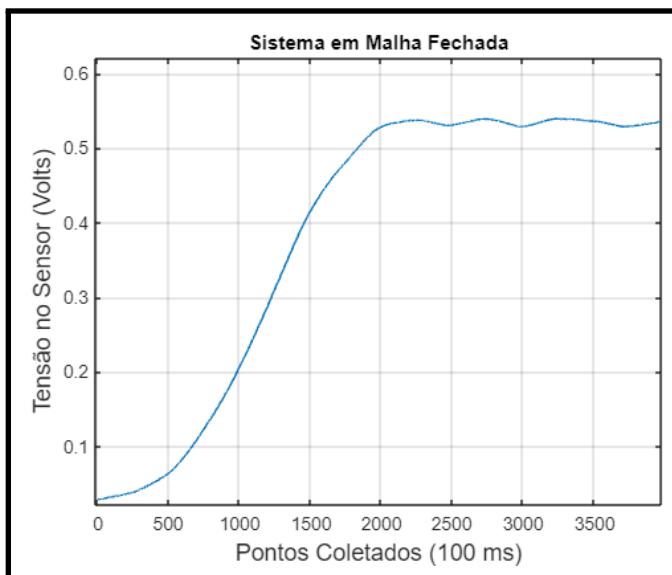


Figura 27. Resultado prático obtido do sistema operando em malha fechada.

Para tratar dos dados obtidos com a simulação prática do sistema foi utilizado o seguinte algoritmo no *Matlab*:

```
% Importa os dados do teste prático.
data_teste_8 = importdata('teste.txt');
valor = data_teste_8(:, 1);
temp = 0:100e-3:length(valor);
% Plota o gráfico aplicando um filtro sobre os valores.
plot(movmean(valor, 1000))
grid on
title ('Sistema em Malha Fechada');
xlabel ('Pontos Coletados (100 ms)', 'FontSize', 13);
ylabel ('Tensão no Sensor (Volts)', 'FontSize', 13);
```

Com a imagem abaixo podemos ver um comparativo entre as duas curvas obtidas, as quais são do ensaio prático realizado e a simulação no *software*.

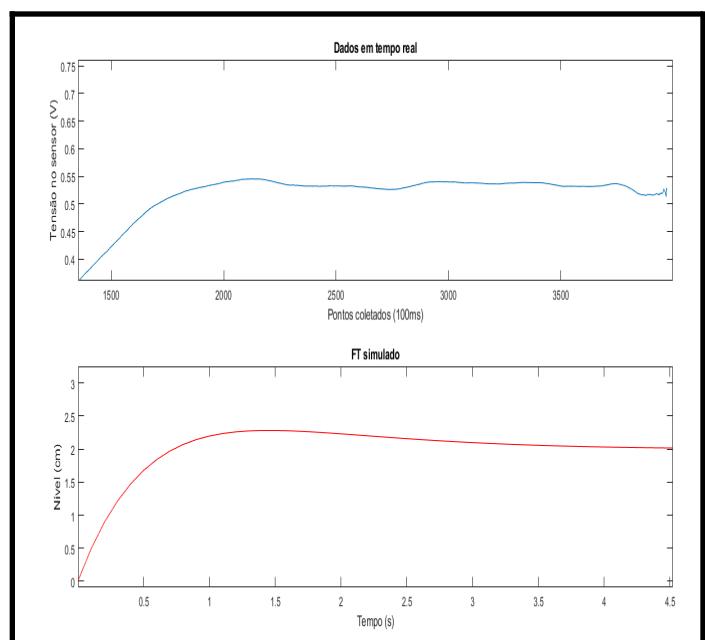


Figura 28. Comparativo dos resultados práticos (obtidos pelo ensaio) e teóricos (obtido pela simulação no *software*) do sistema operando em malha fechada.

## VIII. PROBLEMAS E SOLUÇÕES

Os problemas encontrados durante os testes e no decorrer de todo o projeto foram elencados e diluídos ao longo do presente trabalho. Dessa forma, foi possível seguir uma linha de entendimento melhor e mais sequencial do desenrolar do projeto. Abaixo encontram-se os principais empecilhos que acometeram o projeto e que, de certo modo, tornou mais penoso o seu andamento.

### A. Problemas encontrados no sensor

Inicialmente foi realizada a montagem de um sensor utilizando um eixo com uma boia em uma extremidade, e acoplado ao potenciômetro na outra extremidade. Ao tentar retirar a função de transferência do sensor, observou-se que conforme o nível

do tanque era alterado, um ângulo correspondente era indicado no sensor em forma de tensão, assim, apresentando um funcionamento não linear.

Devido ao comportamento não linear da montagem do sensor com a bóia, foi realizada uma arguição em sala de aula junto com os colegas e o professor. Dessa forma, conseguimos observar que haveria a possibilidade de linearizar o sensor através da montagem de um mecanismo que tornasse o movimento da bóia totalmente linear conforme o nível do tanque subia. Em grupo, pensamos e montamos um mecanismo constituído de uma haste ligada à esfera de isopor (bóia) que, por sua vez, é ligada a um pino e pode se movimentar livremente. Neste pino é ligado uma segunda haste que é fixada no eixo do potenciômetro. Desta forma, a boia faria um movimento fixo de subir e descer conforme o nível e não um ângulo como na configuração anterior do sensor.

#### B. Problemas encontrados na Função de Transferência da planta

Como obtivemos uma simulação em que os valores apresentavam vários picos espúrios, era necessário o tratamento do sinal para conseguirmos uma curva mais condizente com o simulado. Para isso, foi utilizada a função do Matlab “smooth”, que suaviza os dados de resposta no vetor usando um filtro de média móvel. O gráfico correspondente ficou com melhor resposta do que anteriormente.

Após mais algumas tentativas em tratar a curva obtida pelo ensaio da planta, foi utilizada uma outra função no *software* em questão que retornou um resultado melhor do que o obtido no gráfico anterior. A função usada foi a “smoothdata2” que tem como parâmetros de entrada o gráfico em duas dimensões e o método de suavização que se deseja utilizar.

A mudança que houve no algoritmo para essa implementação foi somente na linha que diz respeito à plotagem do gráfico. Onde tínhamos “*plot(x, smooth(y))*” passou a ser “*plot(x, smoothdata2(y, “sgolay”))*”. Após isso, foi obtido o gráfico abaixo, o qual representa o ensaio feito em comparação com a simulação em *software*.

#### C. Problemas encontrados na montagem do circuito

Na montagem do *driver* de acionamento do motor foi observado um elevado aquecimento do transistor usado. Devido a essa potência dissipada pelo transistor TIP142, foi inserido um dissipador de calor em seu encapsulamento para evitar o superaquecimento e possíveis danos ao componente.

## IX. CONCLUSÕES

De forma satisfatória, é possível dizer que foi conseguido realizar o controle de nível proposto no início do projeto. Foram obtidas boas respostas do sistema, tanto operando em malha aberta como em malha fechada. Esses resultados ficaram bem próximos das simulações realizadas no *software*.

## X. REFERÊNCIAS

- [1] OGATA, Katsuhiko. **Modern Control Engineering**, 4th ed, Pearson Education (Singapore) Pvt. Ltd., New Delhi, 2005.
- [2] OGATA, Katsuhiko. **Discrete-time control systems**. 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1995.
- [3] SHANKAR, Ramamurti.. **Fundamentals of Physics I: Mechanics, Relativity, and Thermodynamics**, Expanded Edition. Germany, Yale University Press, 2019.,