



JEAN PAULO DE ALVARENGA - 201621424

Trabalho GAT125 – Laboratório Integrador

**OBTENÇÃO DE MODELO E PROJETO DE CONTROLADOR
DE NÍVEL DE REAGENTES EM PROCESSO QUÍMICO
INDUSTRIAL**

**LAVRAS
2022**

Sumário

1	Resumo	3
2	Objetivo	4
3	Metodologia.....	4
4	Cronograma semanal de atividades	14
5	Validação	15
6	Referências	15

1 Resumo

Em processos industriais o controle de nível de reagentes é de suma importância, para que a mistura ocorra de forma homogênea, gradual, controlada e para que não haja desperdício de reagentes químicos. PERLINGEIRO

O controlador de nível, tem a função de controlar o nível de um reagente em um processo Industrial, em que é necessário que a mistura ocorra de forma homogênea e de acordo com a adição de outro reagente. Composto pelo sistema supervisório (serial do Arduino), o qual um profissional da indústria o administra e define a o nível (*setpoint*) do reagente para o processo.

Durante este projeto foi desenvolvido um sistema de controle reduzido para entender e aplicar os conceitos aprendidos no curso até agora. Desta forma no sistema reduzido uma bomba hidráulica veicular é responsável por encher um tanque a um nível indicado pelo operador, mantendo o nível constante perante a retirada ou a distúrbios externos no líquido. A programação do sistema foi implementada no *software* Arduino em C++, a modelagem e sintonia do controlador no Matlab e a placa de aquisição de dados utilizada foi o Arduino.

Palavras chave: Reagentes químicos, Controle de nível, mistura controlada.

2 Objetivo

O objetivo deste projeto é obtenção de modelos matemáticos a partir da obtenção de dados de uma planta de controle de nível em malha aberta. Com a obtenção da função de transferência é feita a sintonia do controlador PI para controlar a planta em malha fechada. Os objetivos podem ser listados em:

- Obtenção a partir de equações e variáveis físicas de um modelo linear do sistema;
- Projeto da planta em malha aberta;
- Aquisição de dados em malha aberta;
- Projeto dos controladores PI;
- Desenvolvimento e realização de testes e análises;
- Teste de diferentes modelos de função de transferência a fim de representar melhor o sistema em operação.
- Validação do controlador.

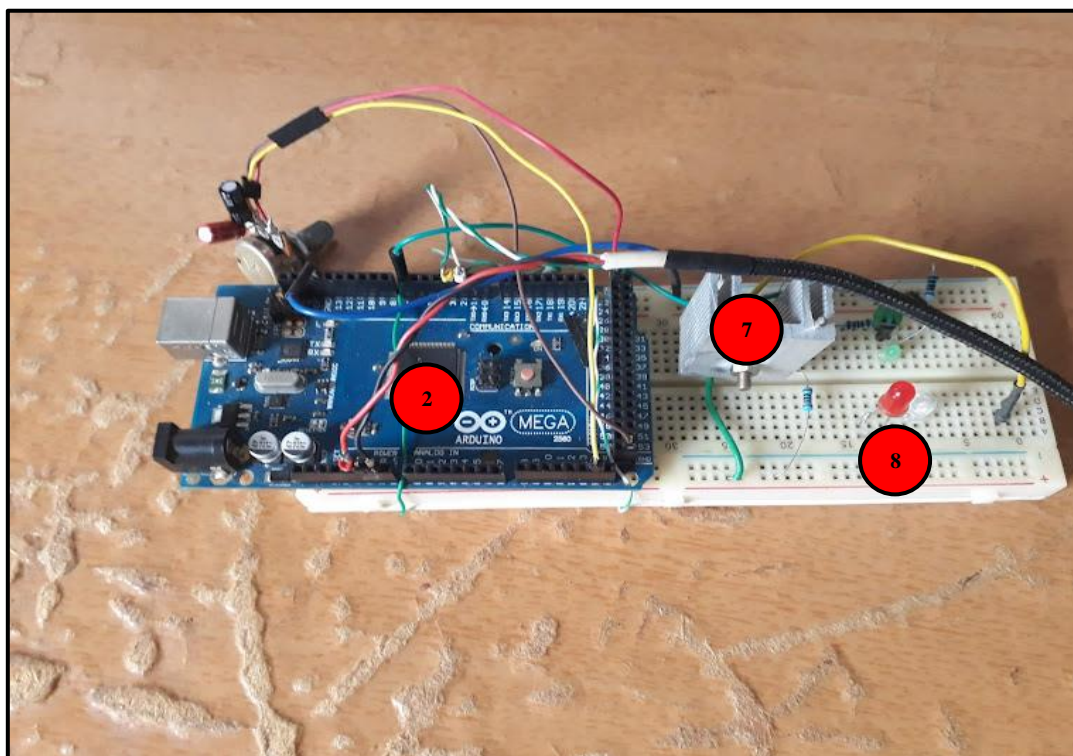
3 Metodologia

3.1 Materiais

O sistema é composto pelos seguintes materiais, representados na figura 1 e na figura 2:

1. Sensor ultrassônico (hc-sr04);
2. Controlador (Arduino);
3. Bomba hidráulica veicular (12V);
4. Mangueiras;
5. Dois tanques (recipiente de *whey protein*);
6. Suporte de madeira para sistema;
7. Driver de potência com *mosfet*;
8. Protoboard;
9. Fonte de alimentação.

Figura 1. Materiais utilizados



Fonte: Autoria própria

Figura 2.

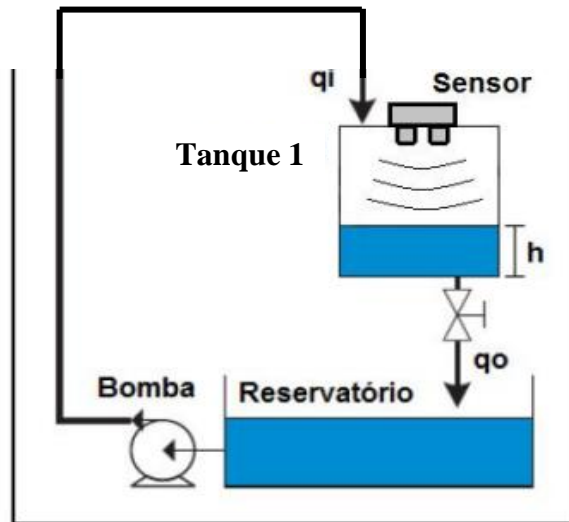


Fonte: Autoria própria

3.2 Modelo conceitual

O modelo conceitual da planta, representado no desenho abaixo, figura 3.

Figura 3.

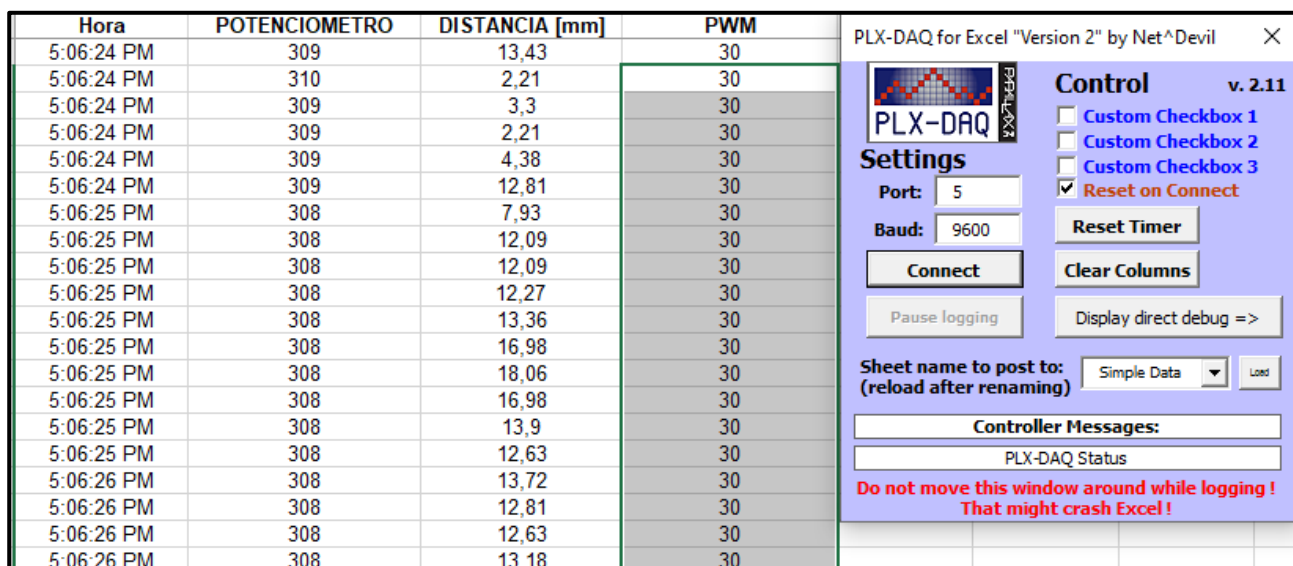


Fonte: <https://www.lume.ufrgs.br/bitstream/handle/10183/150947/001009640.pdf?sequence=1>

3.3 Implementação em Arduino em malha aberta e obtenção de dados via serial.

O desenvolvimento na plataforma Arduino, facilita muita a elaboração de rotinas e de códigos. O sistema foi implementado de forma que um potenciômetro foi utilizado para simular o valor do *duty cycle* a ser enviado para o motor e a partir, desse valor, foram coletados o valor do nível do recipiente em função da altura através do software PLX-DAQ gratuito, conforme a figura 6. A obtenção dos dados em malha aberta foi realizada pelo [código](#).

Figura 6: dados obtidos através do PLX-DAQ em malha aberta.

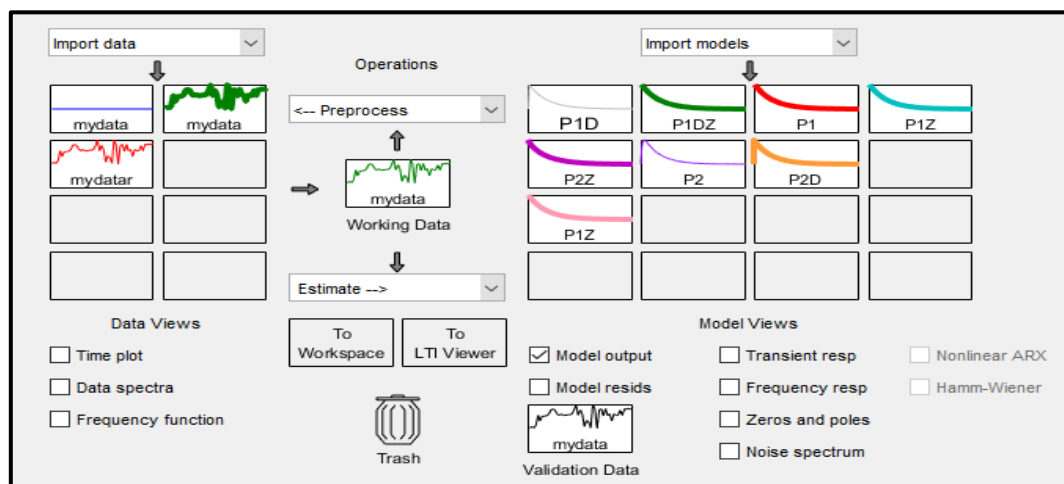


Fonte: Autoria própria

3.4 Projeto do controlador

Os dados gerados pela etapa de leitura da planta em malha aberta foram usados no Matlab para obter diferentes modelos de funções de transferência através do Identificador de sistemas, figura 7.

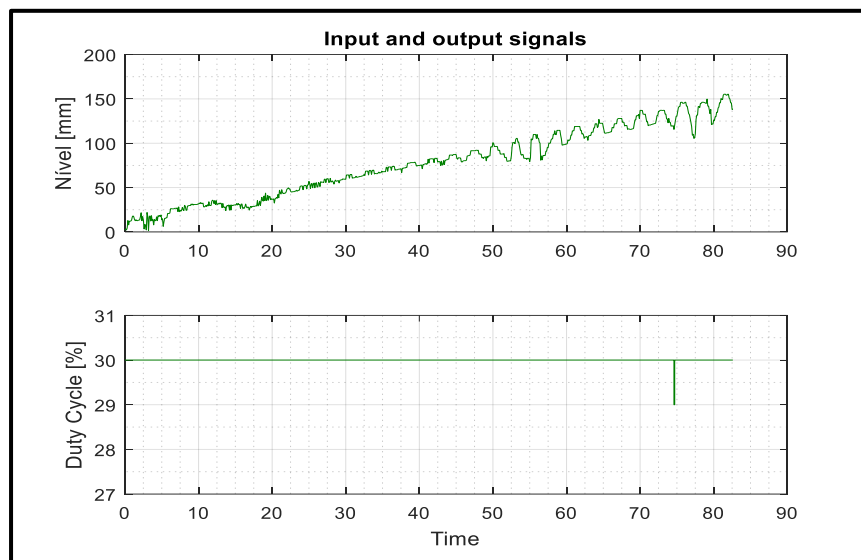
Figura 7: Identificador de sistemas



Fonte: Autoria própria

Os sinais de entrada e de saída foram plotados na Figura 8, de forma que na entrada, $u(t)$ temos o pulso de *duty cycle* de PWM aplicado a bomba hidráulica (atuador) responsável pelo enchimento do tanque, que possui um *range* de 0 a 100%. Para o teste de simulação foi aplicado um degrau de 30% de *duty cycle*, como *setpoint* na bomba e $y(t)$ representa a saída do sensor ultrassônico.

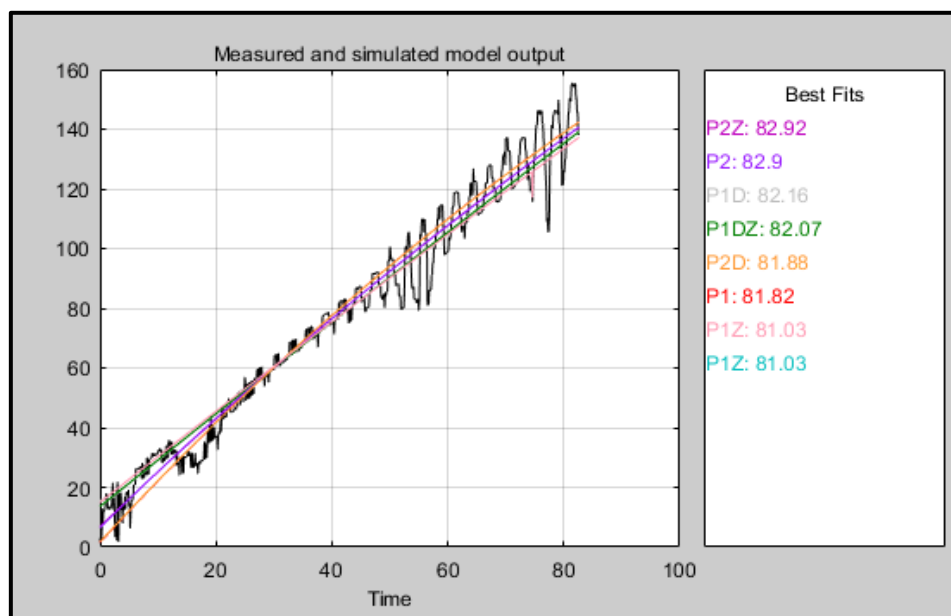
Figura 8: Sinais de entrada e de saída



Fonte: Autoria própria

Aplicando o **ident** (*System identification*) nos sinais de entrada e de saída podemos notar que no modelo onde foi aplicado dois polos e um zero representou melhor a função original do sistema. Outros testes foram feitos com um polo e *delay*, e com apenas dois polos sem *delay* e sem zeros, mas com menor aproximação. Os modelos podem ser observados na figura 9.

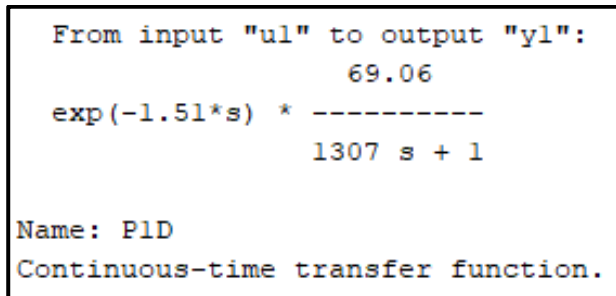
Figura 9: Modelos simulados.



Fonte: Autoria própria

A primeira função de implementada com 1 polo e com atraso de 1,51s utilizando o *autotuning* do Matlab, pode ser visualizada na figura 10, e seus parâmetros de controlador na figura 11 e figura 12.

Figura 10: Função de Transferência H1.



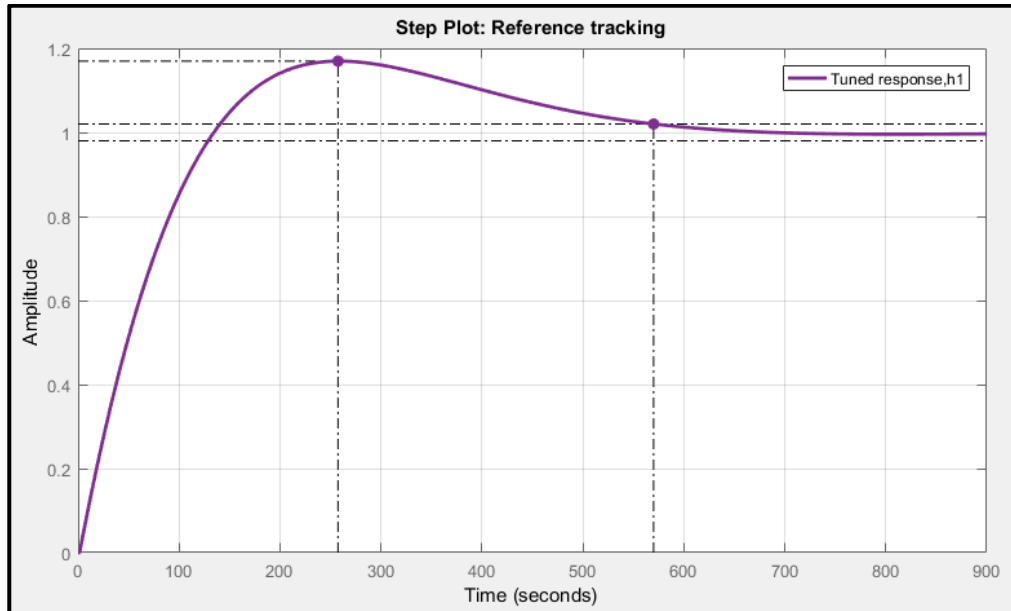
Fonte: Autoria própria

Figura 11: Parâmetros do controlador 1.

Controller Parameters	
	Tuned
Kp	0.2359
Ki	0.0014333
Kd	n/a
Tf	n/a
Performance and Robustness	
	Tuned
Rise time	99.9 seconds
Settling time	570 seconds
Overshoot	17 %
Peak	1.17
Gain margin	38.4 dB @ 1.04 rad/s
Phase margin	68 deg @ 0.0136 rad/s
Closed-loop stability	Stable

Fonte: Autoria própria

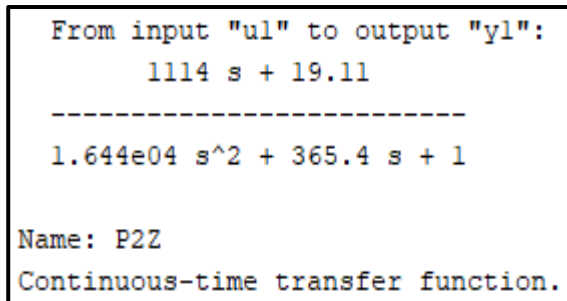
Figura 12: Resposta ao degrau controlador 1.



Fonte: Autoria própria

A segunda função de implementada com 2 polos e sem atraso e com pode ser visualizada na figura 13, e seus parâmetros de controlador na figura 14 e figura 15.

Figura 13: Função de Transferência H3.



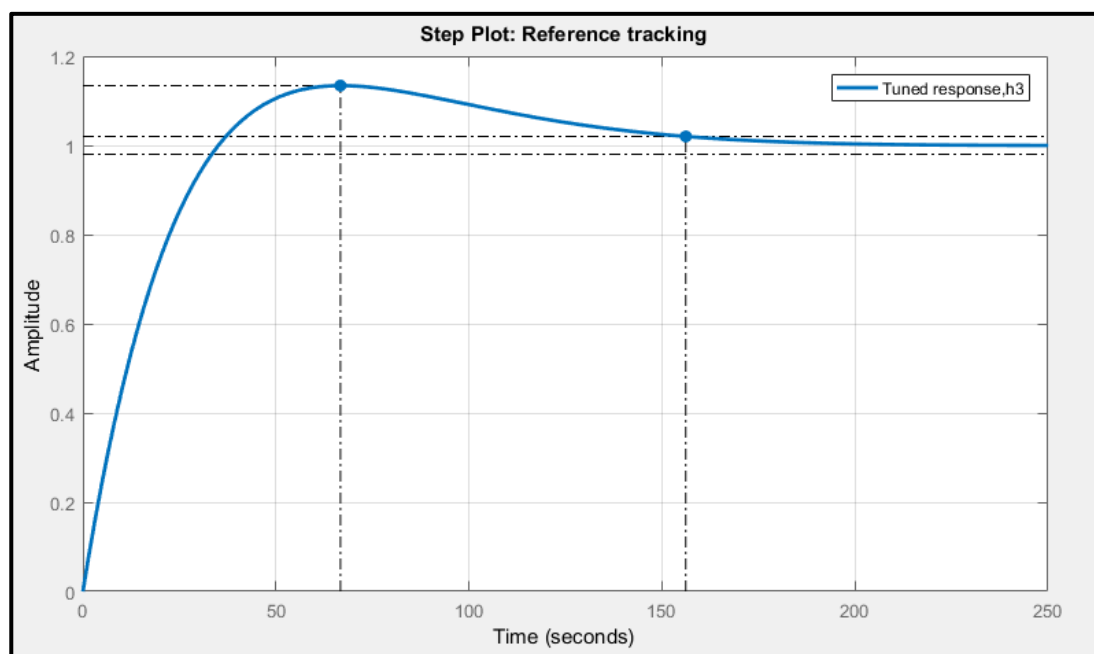
Fonte: Autoria própria

Figura 14: Parâmetros do controlador 2.

Controller Parameters	
	Tuned
Kp	0.79765
Ki	0.017427
Kd	n/a
Tf	n/a
Performance and Robustness	
	Tuned
Rise time	25.8 seconds
Settling time	156 seconds
Overshoot	13.4 %
Peak	1.13
Gain margin	Inf dB @ NaN rad/s
Phase margin	74 deg @ 0.0572 rad/s
Closed-loop stability	Stable

Fonte: Autoria própria

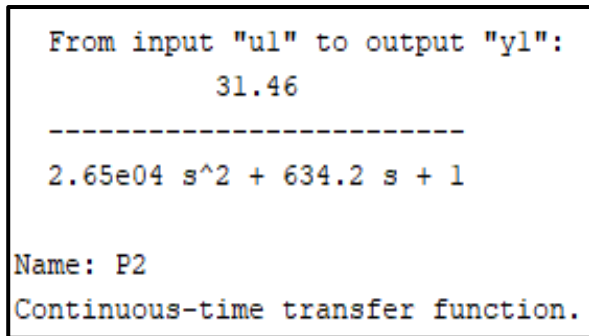
Figura 15: Resposta ao degrau controlador 2.



Fonte: Autoria própria

A terceira função de implementada com 2 polo sem atraso e sem zero pode ser visualizada na figura 16, e seus parâmetros de controlador na figura 17 e figura 18.

Figura 16: Função de Transferência H2.



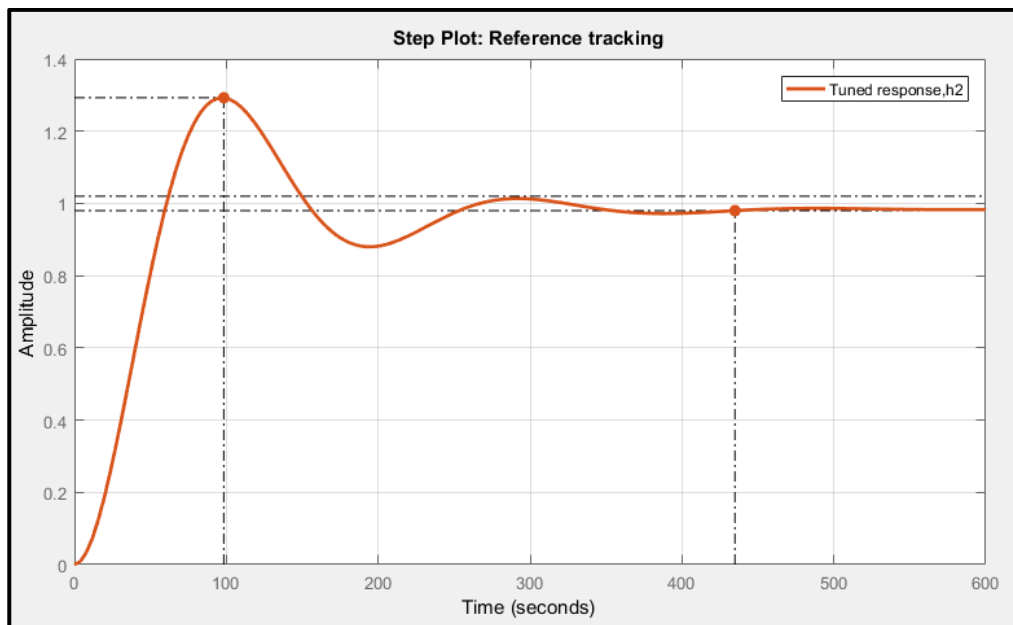
Fonte: Autoria própria

Figura 17: Parâmetros do controlador 3.

Controller Parameters	
	Tuned
Kp	0.97414
Ki	0.00052098
Kd	n/a
Tf	n/a
Performance and Robustness	
	Tuned
Rise time	41 seconds
Settling time	435 seconds
Overshoot	29.3 %
Peak	1.29
Gain margin	Inf dB @ Inf rad/s
Phase margin	38.2 deg @ 0.0305 rad/s
Closed-loop stability	Stable

Fonte: Autoria própria

Figura 18: Resposta ao degrau controlador 3.



Fonte: Autoria própria

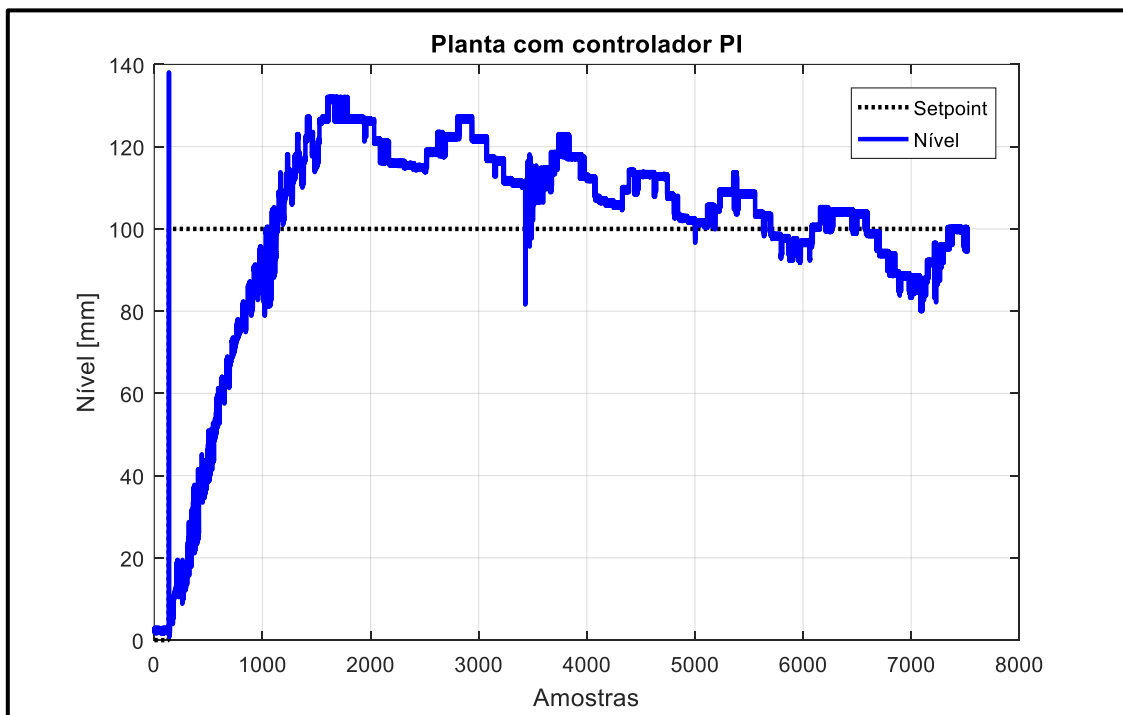
3.5 Implementação no Arduino em malha fechada utilizando o controlador.

A [implementação](#) da planta em malha fechada foi feita utilizando o controlador 2 e o controlador 3, de forma que ambos foram otimizados para haver menor overshoot e menor tempo para alcançar regime permanente, pois nesse tipo de processo o tempo impacta diretamente na produção de produto final.

Controlador 2

Na figura 19 utilizando o controlador 2, foi plotado o nível da água e do *setpoint* em função as amostras coletadas com destaque que o sinal produzido foi muito ruidoso e levando grande tempo para alcançar regime permanente devido a essa condição.

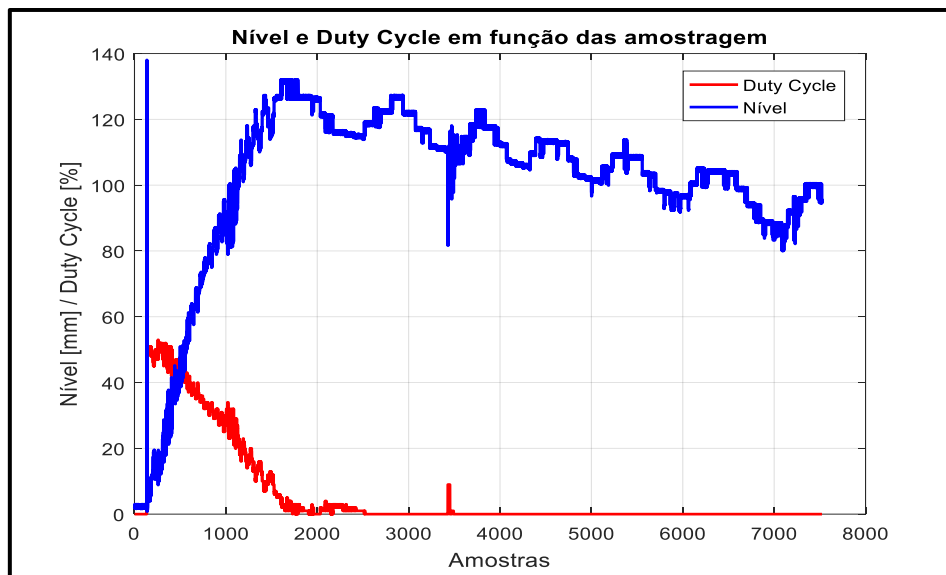
Figura 19: Planta utilizando o controlador.



Fonte: Autoria própria

Na figura 20 temos o nível do fluido e o *duty cycle* na mesma figura

Figura 20: Planta utilizando o controlador 2 em malha fechada.

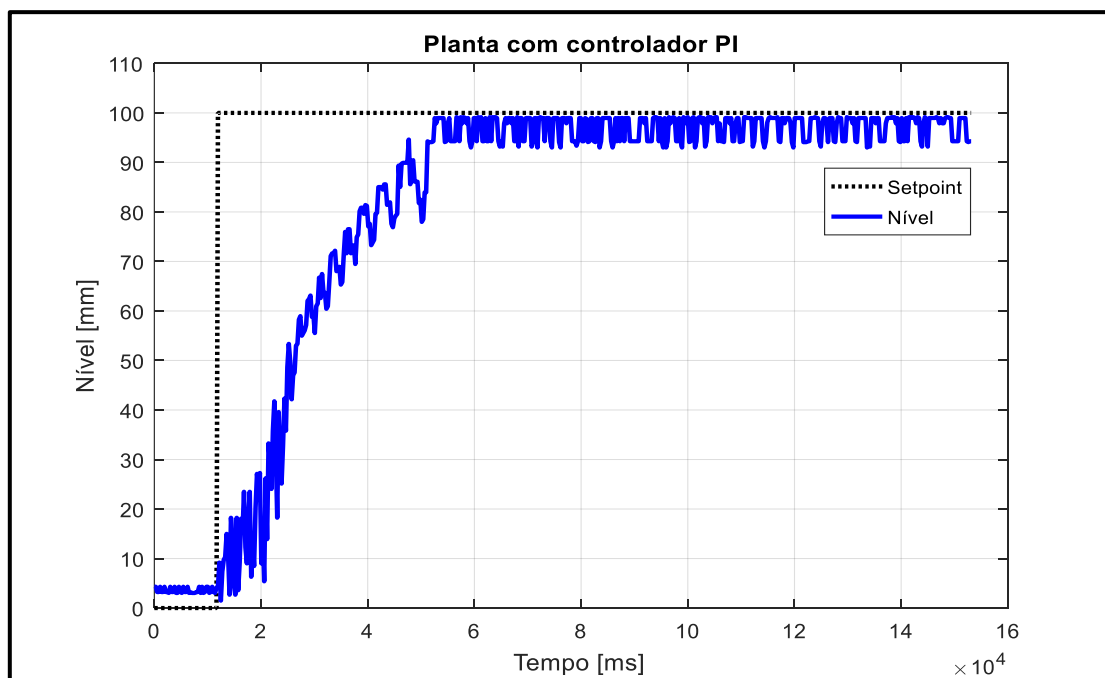


Fonte: Autoria própria

Controlador 3

Na figura 21 utilizando o controlador 3, foi plotado o nível da água e do *setpoint* em função do tempo, em milissegundos.

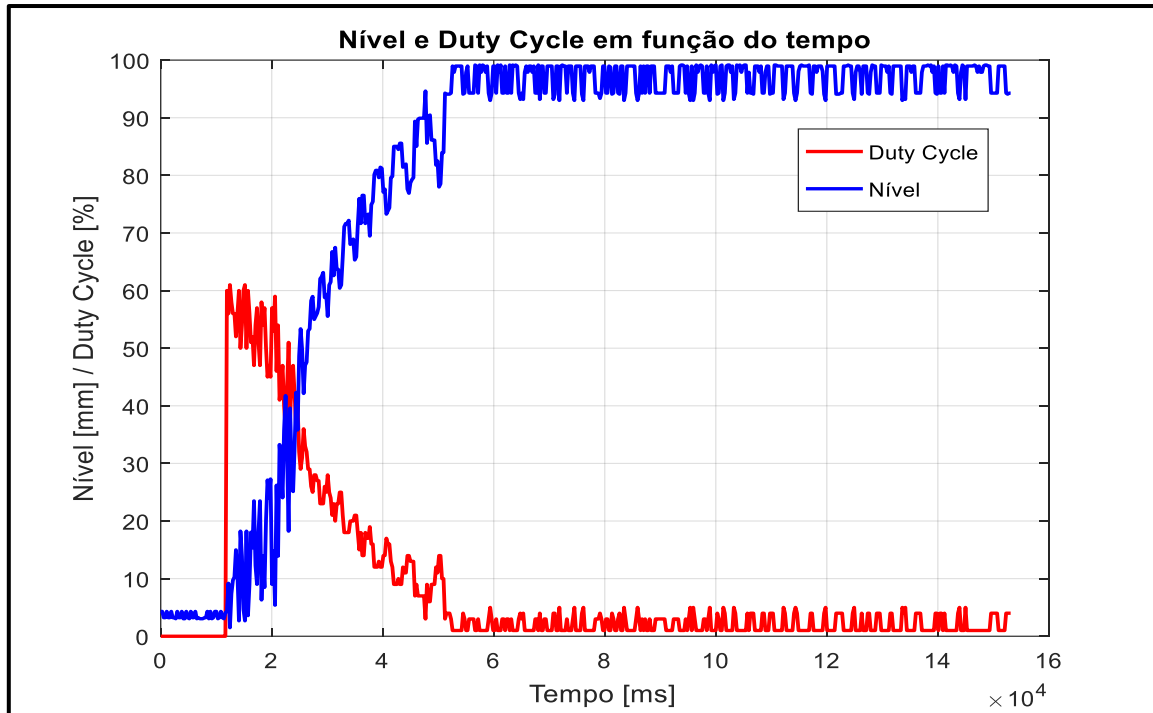
Figura 21: Planta utilizando o controlador.



Fonte: Autoria própria

Na figura 22 temos o nível do fluido e o *duty cycle* na mesma figura.

Figura 22: Planta utilizando o controlador 3 em malha fechada.



Fonte: Autoria própria

Um vídeo da validação do controlador 3 pode ser visualizado no [link](#).

4 Cronograma semanal de atividades

- **Semana 1:** Estudo inicial e modelagem matemática;
- **Semana 2:** Estudo da instrumentação necessária;
- **Semana 3:** Estudo de representação no tempo contínuo;
- **Semana 4:** Estudo de representação no tempo discreto;
- **Semana 5:** Estudo do controle PID;
- **Semana 6:** Projeto e estudo de um controlador PI;
- **Semana 7:** Projeto e estudo de um Controlador por Linearização Exata por Realimentação;
- **Semana 8:** Simulação em malha aberta;
- **Semana 9:** Simulação em malha fechada (realimentação);
- **Semana 10:** Preparação para resultado prático experimental.

- **Semana 11:** Projeto simulação
- **Semana 12:** Projeto simulação
- **Semana 13:** Projeto simulação
- **Semana 14:** Projeto Final
- **Semana 15:** Projeto Final
- **Semana 16:** Projeto Final

5 Validação

Após os testes realizados foi possível compreender que o melhor controlador projetado foi o controlador 3. O controlador apresentou um bom tempo de resposta e diferentemente da simulação quando aplicando a resposta ao degrau, não apresentou overshoot, figura 21, logo se encaixou perfeitamente para a aplicação, onde no misturador, os reagentes químicos também não deveriam ser adicionados sem controle, produzindo uma solução final não homogênea. E o controlador 2 não produziu um bom resultado, pois apresentou alto *overshoot* e dificuldade em se estabilizar (figura 20).

6 Referências

[1] PERLINGEIRO C.A.G. Engenharia de processos: Análise, simulação, otimização e síntese de processos químicos.

[2] Webinar Sistemas de Controle com MATLAB para Iniciantes, site:

<https://www.youtube.com/watch?v=0c7zynjhmPE&ab_channel=OPENCADD> 2019. Acessado 31 de março de 2022.

[3] Desenvolvimento MOSFET driver site:

<<https://electronics.stackexchange.com/questions/267093/using-a-mosfet-driver>> 2022. Acessado 30 de março de 2022.

[4] Controle de velocidade de motor DC com PWM no Arduino UNO site:

<<https://www.filipeflop.com/blog/controle-motor-dc-pwm-arduino-uno/>> 2022. Acessado 30 de março 2022.

[5] Biblioteca sensor Ultrassônico site:

<<https://registry.platformio.org/libraries/ericksimoes/Ultrasonic/examples/UltrasonicSimple/UltrasonicSimple.ino>> 2022. Acessado 4 de abril 2022.

[6] Utilização PLX-DAQ site:

<https://www.youtube.com/watch?v=MBGv77BtDTo&ab_channel=Eletr%C3%B4nicaArduinoeAndroid> 2022 Acessado 8 de abril 2022.

[7] Implementação de controlador no Arduino site:

<<https://gist.github.com/ivanseidel/b1693a3be7bb38ff3b63>> 2022 Acessado 8 de abril 2022.