

Monitoramento de um sistema de controle de nível em planta piloto, utilizando supervisório desenvolvido no software ScadaBR.

Monitoring of a level control system in a pilot plant, using supervision developed in the ScadaBR software.

Fabício Gusmão Ribeiro Silva¹; Davi Leonardo de Souza²

RESUMO

A Indústria 4.0, um conceito que surgiu na Alemanha em 2011, tem se destacado cada vez mais no cenário internacional, mas no Brasil sua implementação tem sido mais lenta. O objetivo principal é desenvolver um supervisório funcional no ScadaBR, conectado ao Arduino Uno, para monitorar e controlar uma planta piloto de controle de nível, tanto em malha aberta quanto em malha fechada com controladores PI. Foi utilizado uma planta piloto composta por um reservatório de líquido, bomba, sensor ultrassônico e Arduino Uno, foram realizadas etapas de medição de nível, conexão com o ScadaBR, acionamento da bomba e implementação de dois controladores PI. Foi realizado testes em malha aberta e em malha fechada, em relação aos testes em malha fechada houve um melhor desempenho com um controlador tipo PI proposto por Pitta et al. (2023) em comparação com outro controlador PI desenvolvido por Almeida et al. (2021). O supervisório desenvolvido mostrou-se eficaz na coleta de dados em tempo real e na implementação de estratégias de controle. Os objetivos do estudo foram alcançados com sucesso, e sugere-se para trabalhos futuros a integração de outros dispositivos e a aplicação de técnicas de machine learning para aprimorar o desempenho do sistema de controle.

Palavras-chave: Indústria 4.0, supervisório com ScadaBR, Arduino Uno, Controladores PID, Solução Econômica.

ABSTRACT

Industry 4.0, a concept that emerged in Germany in 2011, has been increasingly prominent on the international stage, but its implementation in Brazil has been slower. The main objective is to develop a functional supervisory system in ScadaBR, connected to the Arduino Uno, to monitor and control a pilot plant for level control, both in open-loop and closed-loop with PI controllers. A pilot plant consisting of a liquid reservoir, pump, ultrasonic sensor, and Arduino Uno was used, and steps were taken for level measurement, connection to ScadaBR, pump activation, and implementation of two PI controllers. Tests were conducted in open-loop and closed-loop configurations, with better performance observed in closed-loop tests using a PI controller proposed by Pitta et al. (2023) compared to another PI controller developed by Almeida et al. (2021). The developed supervisory system proved effective in real-time data collection and control strategy implementation. The study's objectives were successfully achieved, and for future work, the integration of other devices and the application of machine learning techniques to improve control system performance are suggested.

Keywords: Industry 4.0, Supervisory with ScadaBR, Arduino Uno, PID Controllers, Economic Solution.

¹ Aluno do Curso de Graduação em Engenharia Química, Universidade Federal do Triângulo Mineiro, Uberaba, Minas Gerais, Brasil.

E-mail: fabriciogus1@gmail.com

ORCID: <https://orcid.org/0000-0002-5881-4133>

² Professor, Universidade Federal do Triângulo Mineiro, Uberaba, Minas Gerais, Brasil.

E-mail: davi.souza@uftm.edu.br

ORCID: <https://orcid.org/0000-0002-1995-9057>

1. INTRODUÇÃO

O termo de indústria 4.0 foi citado na Alemanha pela primeira vez em 2011, e desde então é um tema que vem cada vez mais tomando frente as indústrias internacionais e nacionais, e auxiliando-as a dar um norte as estruturas fabris. (PINTO, 2020). No Brasil, as implementações relacionadas a indústria 4.0, vem caminhando lentamente segundo Pinto (2020), se compararmos aos países europeus e norte-americanos.

Ao analisar as definições e princípios da indústria 4.0, SANTOS et al. (2018) define bem alguns deles:

- Capacidade de operação em tempo real: trata-se da obtenção e tratamento de dados de forma rápida e eficaz, podendo tomar decisões em tempo real dos processos;
- Virtualização: em resumo é a existência de cópias virtuais das fábricas inteligentes, permitindo o monitoramento a distância de todos os processos por meio de sensores, controladores, e todos os acessórios que compõem uma planta automatizada;
- Descentralização: de acordo com a produção em tempo real, as tomadas de decisões podem vir a ser tomadas por um sistema cyber-físico;
- Orientação a serviços: é definido como a utilização de arquiteturas e softwares orientadas a serviços;
- Modularidade: módulos de produção que oferecem flexibilidade para as máquinas de acordo com a demanda, podendo ser acoplados ou desacoplados.

Em ambientes fabris, é muito comum, ter-se sistemas de supervisórios para acompanhamento e modulação de processos, pois em muitos casos se torna complexo o controle de variáveis, como temperatura, pressão, vazão, nível entre outros, de forma manual.

Quando se fala em capacidade de operação em tempo real, há diversas variáveis a serem controladas, como citado anteriormente, e para padronizar os valores destas variáveis, e mantê-las sob um padrão pré-estabelecido, é muito comum a utilização de controladores, sejam eles do tipo PI (Proporcional-Integrativo), PD (Proporcional-Derivativo), ou PID (Proporcional Integral Derivativo).

A ação proporcional (P) atua no ganho do controlador, a ação integral (I) é ligada à precisão da resposta do sistema, sendo assim, é baseada no tempo em que o erro ocorre,

já a ação derivativa (D) auxilia no ajuste da variável, ou seja, atuando na taxa de variação do erro (SWIECH et al., 2004).

Junto aos controladores, é muito usual a sua utilização em conjunto com supervisórios, que são sistemas SCADA (Sistemas de Supervisão, Controle e Aquisição de Dados). Esses sistemas automatizam e facilitam o controle e acompanhamento do processo (MIRANDA, 2021). Em geral são interfaces amigáveis, que se comunicam com sensores e controladores, em sua maioria, que ficam na área da produção, e captam dados, e replicam em suas telas todos que estão aptos a serem adquiridos em tempo real.

De acordo Kazala e Straczynski (2019) apud Miranda (2021), o uso de softwares gratuitos não é algo comum no setor industrial. Dessa forma, quando se trata de automação de projetos de pequenos portes, devido ao custo bastante elevado dessa ferramenta, é comum utilizar softwares SCADA com licenças abertas.

O ScadaBR é uma opção de software de criação de supervisorio e aquisição de dados livre e gratuito, e de código-fonte aberto para o desenvolvimento de aplicações de engenharia (MIRANDA, 2021). A utilização do ScadaBR no desenvolvimento e simulação de processos químicos, oferece uma gama de oportunidade de estudar o controle e funcionalidade de equipamentos de processos industriais, sem a necessidade de sua compra (CARDOSO, 2022), abrindo oportunidades de realizar estudos em projetos pilotos de baixo orçamento.

A comunicação do processo ao supervisorio em plantas industriais, geralmente é realizada por algum elemento intermediário, o mais comum são CLPs (Controladores Lógicos Programáveis) e/ou UTRs (Unidades de Terminal Remotas). Essas estações remotas, como são chamadas, são unidades computacionais específicas, que tem como funcionalidade ler entradas, realizar cálculos e/ou controles, e atualizar saídas (CAVALCANTI, 2008).

Para projetos pequenos é comumente utilizado microcontroladores programáveis, que é em resumo um computador de um único chip, que tem diversas funções, como memória, periféricos de entrada e saída, temporizadores, dispositivos de comunicação serial, entre outros. Quando se fala em microcontroladores, as placas Arduino tornam-se bastante presentes no cenário de sistemas de controle, e em muitos casos, é bastante eficiente e economicamente rentável (GIMENES; PEREIRA, 2015).

Projetos pequenos e a baixo custo, permitem muitas vezes, simular a realidade de uma planta industrial real, com dados e variáveis reais de processo, sejam elas, nível, temperatura, pressão, entre outras inúmeras. As placas Arduino realizam uma função essencial que é de comunicação entre o processo e os supervisórios (SCADA), e trabalham como CLPs, se comparar a uma escala industrial.

Com o objetivo de realizar uma coleta de dados e análises em tempo real de uma planta piloto de controle de nível, criou-se um supervisorio no software ScadaBR, que com o auxílio de um Arduino Uno, realizará a comunicação e controle do processo utilizando controladores já pré-definidos. Os controladores que serão testados foram elaborados por Almeida et al. (2021) e Pitta et al. (2023).

2. MATERIAIS E MÉTODOS

Projeto de Controladores

Os controladores são dispositivos que mantém a variável de processo em um valor pré-determinado. É um instrumento que faz uma análise, entre o valor da variável de processo com o *setpoint* (SP) e exerce uma ação corretiva na variável manipulada (MV) (BEGA, 2011 apud SOARES, 2018).

Os controladores PID têm três modos de atuação o modo de ação proporcional, ação integral e derivativa.

Com modo de ação proporcional (P), os controladores do tipo P são responsáveis por tomar uma ação proporcional ao desvio do *setpoint*, isto é, o quão mais longe a variável controlada se distanciar do valor estabelecido (*setpoint*) mais robusta é a ação do controlador, e quanto mais próximo do valor desejado, mais amistosa é sua ação (SEBORG et al., 2011).

A ação integral (I) tem como principal função a eliminação do *offset*, que é uma das desvantagens do controle proporcional. Esse *offset* é a distância entre o *setpoint* e o valor medido da variável controlada em um sistema de controle (SEBORG et al., 2011).

Já a ação derivativa (D), vem como principal intuito de amenizar um comportamento oscilatório que pode ocorrer na variável controlada (SEBORG et al., 2011).

O princípio no projeto de um controlador PID é a determinação da função de transferência (FT) que permite encontra informações para o procedimento do projeto de controlador, e auxilia no estudo da dinâmica do processo. O modelo da função de

transferência caracteriza a relação dinâmica de duas variáveis do processo, uma independente (variável de entrada) e outra dependente (variável de saída). A dependente, ou controlada, como pode ser chamada, é a que tem como principal função manter o valor desejado, já a independente, ou manipulada, auxilia o processo para manter a variável controlada no valor de *setpoint* (SEBORG et al., 2011).

As funções de transferências podem apresentar diversos equacionamentos, que vão depender principalmente do processo em que se está interessado, mas as mais comuns são as de primeira e segunda ordem com tempo morto. As **Equações 1 e 2** demonstram suas formas:

$$G(s) = \frac{K e^{-\theta s}}{\tau s + 1} \quad (1)$$

$$G(s) = \frac{K e^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (2)$$

O termo K representa o ganho do processo, o τ a constante do tempo, e a mesma informa da velocidade e características da resposta do sistema, e por fim, o θ representa o tempo morto.

A obtenção desses parâmetros é de extrema importância no projeto de controladores, e uma das formas mais utilizadas é a de curva de reação. A curva de reação é obtida retirando dados da variável de saída (controlada) em função do tempo.

A **Figura 1** exemplifica dois tipos de curva de reação, um para processo integrante, e outro para processo auto regulante após ocasionar uma pequena mudança na saída do processo, que muitas vezes é apenas uma mudança de *setpoint*.

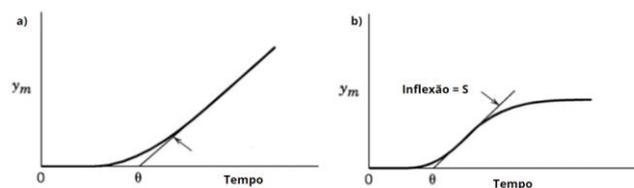


Figura 1. Curva de reação de um sistema integrante (a) e auto regulante (b)

Fonte: Adaptado de SEBORG et al. (2011).

A **Figura 1** demonstra dois gráficos, sendo **a)** para o integrante que em termos de processo é um sistema no qual a resposta a uma perturbação na entrada do sistema, irá

ocasionar continuamente um aumento ou redução na saída do sistema sem nunca atingir um estado estável.

A parte **b)** da **Figura 1** indica um sistema auto regulante, que é o comum de vários processos, que em de forma resumida é capaz de regular sua própria saída para compensar mudanças nas condições operacionais.

Nas duas figuras, é demonstrado a tangente no ponto de inflexão sendo traçada na curva de reação para a obtenção do tempo morto (θ) em função dos dados da variável de saída (y_m) em função do tempo.

A determinação gráfica dos parâmetros K , τ e θ é demonstrada detalhadamente pela **Figura 2**.

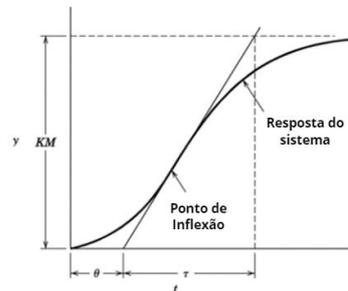


Figura 2. Obtenção dos parâmetros pela curva de reação.

Fonte: Adaptado de SEBORG et al. (2011).

Após traçado a tangente pelo ponto de inflexão (*Inflection point*), é possível identificar o tempo morto (θ) do processo, e consequentemente com as linhas pontilhadas determinar o valor do tau (τ) que é a diferença entre o tempo (t) indicado no eixo das abcissas pelo valor de tempo morto. No eixo das ordenadas, o valor de y é o produto do ganho do processo (K) pela perturbação de amplitude M na variável manipulada (SEBORG et al., 2011).

De posse das informações de processo necessárias (K , τ e θ), pode-se projetar um controlador PID que irá tomar ações baseado em seus parâmetros de operação com a finalidade de reduzir o erro e controlar o sistema de forma estável e eficaz. Os parâmetros para o funcionamento de um controlador PID são demonstrados pela **Equação 3**:

$$G_C(s) = K_C \left(1 + \frac{1}{\tau_I s} + \tau_D s \right) \quad (3)$$

Na **Equação 3**, o termo K_c representa o ganho da função no que diz respeito a parte proporcional do controlador (P). Já o termo τ_I representa a constante de tempo relacionada a parte integral do controlador, e τ_D a parte derivativa.

Ao combinar os parâmetros e projetar o controlador PID, pode-se esperar diversas respostas no que diz respeito ao controle do processo em caso de perturbações do sistema. Ao final, na maioria das vezes é necessário um ajuste fino dos parâmetros em questão, mas há respostas padrões que são esperadas de controladores PID. Esses comportamentos podem ser observados na **Figura 3**.

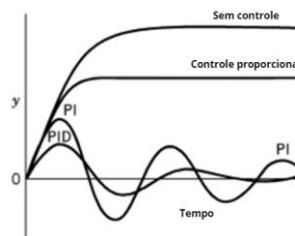


Figura 3. Comportamento de processos com controladores PID.

Fonte: Adaptado de SEBORG et al. (2011).

O comportamento bem demonstrado na **Figura 3** reflete as ações dos controladores no processo. Um processo sem controle (*no control*) está longe do *setpoint* estipulado. Ao inserir um controle proporcional (P), há um ganho no sistema (K_c) que faz com que há uma resposta mais rápida, porém ainda fica distante do valor desejado dependendo do valor atribuído à variável (SEBORG et al., 2011). A **Figura 4** demonstra o comportamento do sistema segundo a variação de K_c .

Ao inserir a parte integral do controlador (τ_I) tem-se um controlador do tipo PI, e é perceptível que o processo, ao analisar a **Figura 3**, tende a buscar de forma mais robusta o *setpoint*, porém é gerado oscilações dependendo do seu valor, e a resposta do mesmo é mais lenta se comparado ao do tipo P. A principal função da parte integral é eliminar o *offset* (distância entre o *setpoint* e $y(t)$ em um sistema de controle) como é notado pela **Figura 4 – b)** (SEBORG et al., 2011).

A parte derivativa (τ_D) entra no processo com a intuição de reduzir a amplitude de oscilação como, pode ser analisado pela **Figura 4 – c)**. Mas, como as demais variáveis, é necessário analisar a resposta do processo ao projetar um controlador PID, e na maioria dos casos, será necessário um ajuste fino das variáveis para se adequar ao processo. (SEBORG et al., 2011).

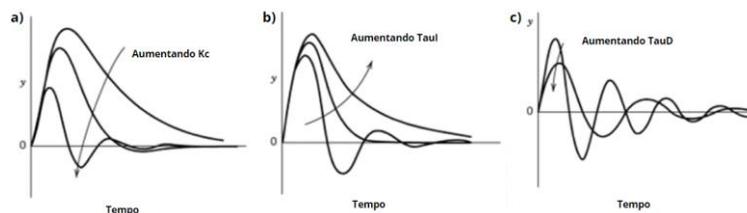


Figura 4. Comportamento da parte proporcional (a), derivativa (b) e integral (c) de um controlador PID.

Fonte: Adaptado de SEBORG et al. (2011).

Arduino

A Plataforma Arduino é um *software* que realiza o desenvolvimento de projetos elaborados em sua própria linguagem de programação baseado em C/C++. Existem diversos modelos de placas Arduino no mercado atual, e fazem um enorme sucesso, pois são fáceis de se manipularem. Seu *hardware* e *software*, são *open-source*, que permite sua aplicação e modificação por infinitas vezes. Na internet, encontra-se diversos fóruns, comunidades e inúmeras aplicações a muitas situações, e muitas as vezes são disponibilizadas pela própria comunidade. (GIMENES; PEREIRA, 2015).

Como comentado, existem diversos modelos, mas o enfoque deste projeto é no Arduino Uno, que é uma placa baseada no microcontrolador ATmega328, e que permite a automação de projetos de diversos segmentos. Ele possui 14 I/Os digitais e 6 entradas analógicas, além de uma comunicação serial UARTTTL (5V), e comunicação serial através de porta USB, que é que foi utilizada no projeto para conectar-se ao supervisorio.

O código e a lógica de programação foram desenvolvidos utilizando o *software* Arduino IDE.

No trabalho foi utilizada a unidade experimental para coleta de dados da variável controlada (vazão da bomba) pelo tempo, que é representada pela **Figura 5**:

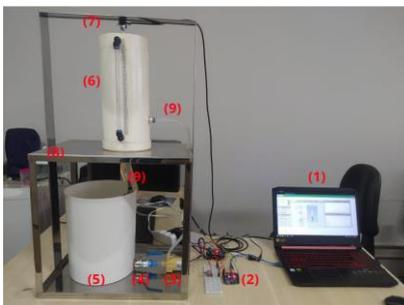


Figura 5. Unidade experimental de controle de Nível.

Fonte: do Autor (2024).

1. Computador e Supervisorio ScadaBR;
2. Microprocessador (Arduino UNO);
3. Bomba (elemento final de controle);
4. Válvula esférica manual;
5. Reservatório de líquido recirculante;

Reservatório para o controle de nível de líquido;

6. Medidor de nível baseado em sensor ultrassônico;
7. Suporte metálico para os reservatórios;
8. Mangueiras para condução de líquido

Inicialmente realizou-se a programação no *software* Arduino IDE que teve como objetivo principal fazer a conexão entre a planta piloto e o ScadaBR. Indo mais a fundo na programação, pode-se separar ela em quatro etapas principais que condizem em uma sequência lógica:

- Etapa de medição de nível;
- Etapa de conexão ScadaBR;
- Etapa de acionamento da bomba;
- Etapa do controlador PID.

Etapa de medição de nível

Inicialmente é necessário medir o nível do tanque indicado pelo número 6 presente na **Figura 5** através do sensor (número 7) a fim de averiguar seu estado inicial. Para fazer essa medição foi utilizada a biblioteca *NewPing.h* que é responsável por habilitar e controlar a leitura do sensor ultrassônico utilizado no presente trabalho.

Esta etapa consiste em indicar para o sensor ultrassônico realizar uma medição do nível do tanque.

A parte principal da programação que é responsável por essa medição de nível se encontra na **Tabela 1**.

Tabela 1. Programação de medição de nível – Parte 1.

```
SetPinFrequencySafe(3, frequency);  
medicao = sonar.ping_median(5);  
Nivel = dist - (float)medicao/58.8;  
int32_t Nivel_Scada = (int32_t)(Nivel);  
mb.Hreg(REGN, Nivel_Scada); // envia valor de nível para ScadaBR  
mb.task();
```

Fonte: do Autor (2024).

Etapa de conexão ScadaBR

A partir dos dados obtidos pelo sonar, utilizando a biblioteca *ModbusRTU.h*, é realizado o envio dos dados do ultrassônico para o ScadaBR.

Em primeiro momento é necessário que o ScadaBR enxergue o Arduino Uno, e para isso identifica-se o mesmo como escravo e coloca-se uma numeração que auxilia o ScadaBR a identificar o hardware utilizando *#define SLAVE ID 1*.

Utilizou-se a função *#define* (**Tabela 2**), à medida que foi necessário enviar ou receber dados do supervisorio durante a programação.

Tabela 2. Programação de medição de nível – Parte 2.

```
//Definindo o ID do Escravo para o ScadaBR indentificar os pontos de conexões com Arduino Uno
#define SLAVE_ID 1
#define REGN 2
#define DEGRAU_VALOR 3
#define POT_VALOR 4
#define BOMBA_VALOR 5
```

Fonte: do Autor (2024).

Etapa de acionamento da bomba

Um dos dados principais que será recebido pelo Arduino Uno (número 2), a partir de valores inseridos diretamente no ScadaBR, é a vazão da bomba. Desta forma, foi inserido na tela do supervisorio um campo para a inserção deste valor, que posteriormente será enviado para a bomba indicada pelo número 3 na **Figura 5**.

De posse do valor da vazão desejado, a bomba responderá o comando e enviará uma vazão de água para o reservatório de líquido recirculante.

Etapa do controlador PID

Para implementar o controlador PID na programação foi necessário inserir a estrutura *If* para durante a interação no supervisorio criado no ScadaBR, o usuário possa alternar entre malha aberta e malha fechada (PID).

A alternância entre as malhas pode ser feita selecionando o botão de “Malha Aberta” ou o de “Malha Fechada” conforme observado na **Figura 6** que representa o supervisorio criado no ScadaBR.

3. RESULTADOS

A coleta de dados da leitura de nível pelo supervisorio elaborado (**Figura 6**) foram obtidos a partir de dois testes realizados na planta piloto, um com o sistema em malha

aberta, e outro em malha fechada utilizando dois controladores PI elaborados por Pitta *et al.* (2023) e Almeida *et al.* (2021).

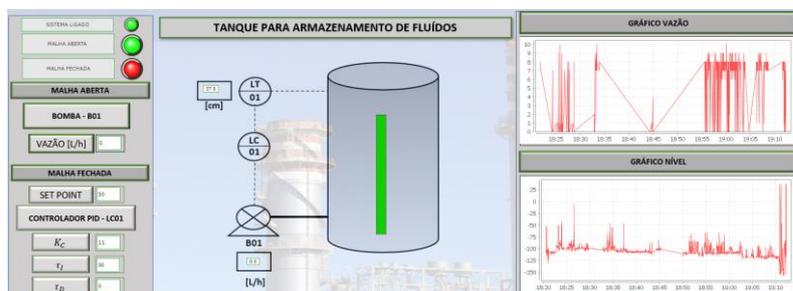


Figura 6. Supervisório criado no ScadaBR.

Fonte: do Autor (2024).

Na parte superior esquerda da **Figura 6** encontra-se os botões ligar/desligar o sistema, e para alternância entre malha aberta e malha fechada. Os locais, logo abaixo, para inserir pulsos em malha aberta (Vazão) e em malha fechada (*Setpoint*). No centro é uma imagem representativa da parte do sistema de nível, a bomba e o sensor ultrassônico conectados aos *data points* do ScadaBR. A esquerda encontra-se os gráficos de vazão e nível do processo.

Durante o primeiro teste em malha aberta, foi estabilizada a planta piloto, e posteriormente realizado um pulso, devido a planta piloto ser regida por um sistema integrante, para obter-se a curva de reação do sistema, que é demonstrada pela **Figura 7**.

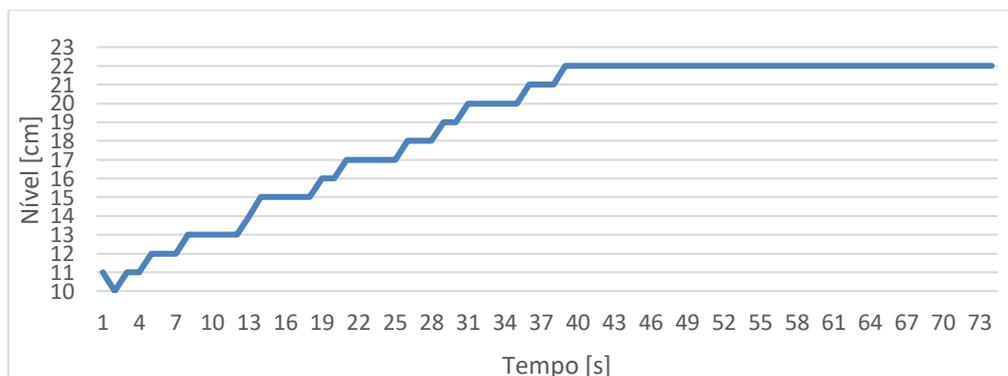


Figura 7. Curva de Reação.

Fonte: do Autor (2024).

A partir da curva de reação, obteve-se a função de transferência representada pela **Equação 4**, indicando que é possível obtê-la através de uma simulação utilizando o supervisório criado.

$$G(s) = \frac{0,091e^{-3s}}{36s + 1} \quad (4)$$

Obteve-se desta forma um ganho do processo de 0,091 (K), um tempo morto (θ) de

3 segundos, e o valor do tau (τ) de 36 segundos.

Após o teste em malha aberta foram realizados dois testes com controladores PI desenvolvidos por métodos diferentes, sendo um mais robusto e outro voltado para performance. Os dois foram testados na planta em questão e os valores das variáveis estão indicados na **Tabela 3**:

Tabela 3. Dados dos Controladores Testados

Teste	Controlador	K_c	τ_i	Autores
1°	PI	30,24	12,87	Almeida et al. (2021)
2°	PI	11	20	Pitta et al. (2023)

Fonte: do Autor (2024).

O 1° teste foi realizado com o controlador elaborado por Almeida et al. (2021), e obteve-se comportamento indicado pela **Figura 8**.

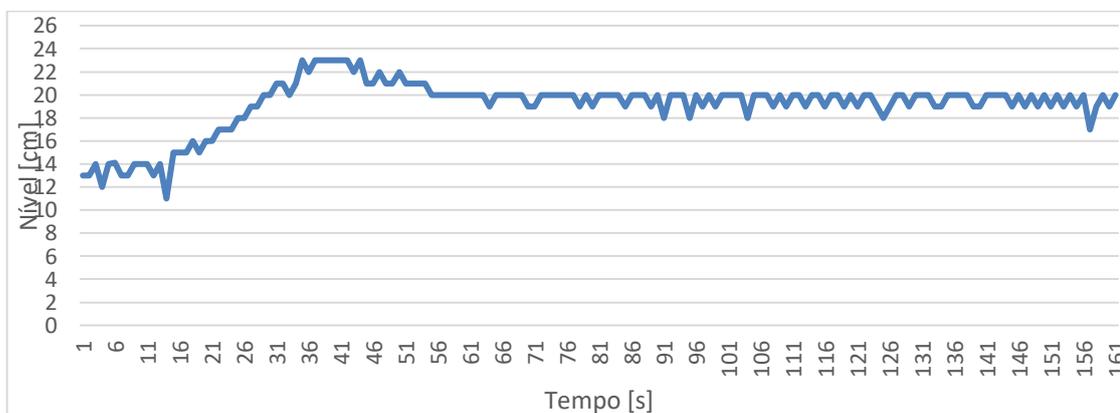


Figura 8. Malha Fechada - Controlador PI - Teste 1.

Fonte: do Autor (2024).

Após estabilizar a planta em 14 cm de nível, foi inserido um valor de *setpoint* de 20 cm de nível. O controlador se comportou da maneira esperada alcançando o valor desejado, entretanto, devido ao seu valor elevado de K_c (30), acabou tendo um *overshoot* no valor de 3 cm após chegar no *setpoint* desejado, e após aproximadamente 25 segundos já se estabilizou no valor alvo de 20 cm.

Já no 2° teste com o controlador de Pitta et al. (2023), em que foi realizado um estudo de otimização para esse mesmo processo, obteve-se um comportamento melhor, conforme pode-se observar pela **Figura 9**:

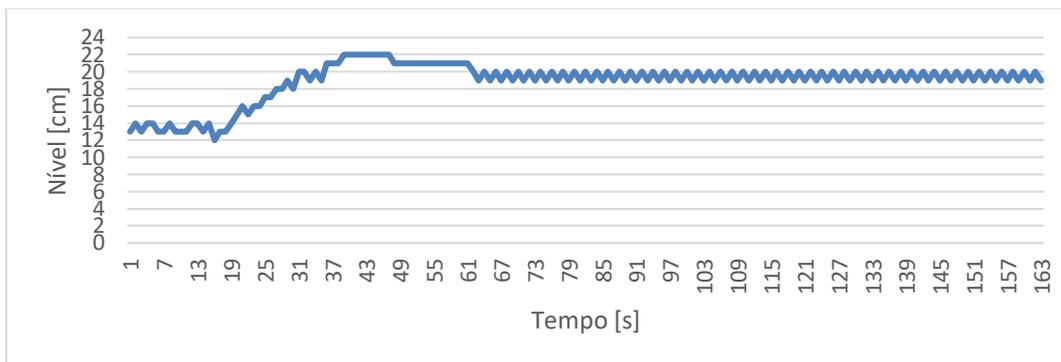


Figura 9. Malha Fechada - Controlador PI - Teste 2.

Fonte: do Autor (2024).

No segundo teste após inserir um pulso para 20 cm de nível, o controlador buscou o *setpoint* conforme esperado e de forma rápida analisando o tempo da chegada ao valor alvo menos tempo em que o sistema se encontrava estabilizado. Diferente do anterior houve um *overshoot* de apenas 1,5 cm, porém com a mesma duração de 25 segundos. É importante destacar que após o atingimento do *setpoint*, o controlador em questão ficou bastante estável atribuído muito ao seu valor da parte integral de 20, que tem por função eliminar o *offset* do sistema.

4. DISCUSSÃO

Em suma, o supervisório desenvolvido demonstrou ser altamente funcional, atendendo com sucesso aos objetivos propostos de monitoramento em tempo real do sistema, apresentando dados de nível e vazão de forma gráfica e operacionalmente eficaz tanto em malha aberta quanto em malha fechada. A interface intuitiva oferece ao usuário a capacidade de alternar entre diferentes configurações e testar controladores PID específicos para o sistema, facilitando o estudo da dinâmica e controle do processo.

Os testes em malha aberta forneceram uma função de transferência satisfatória, fundamental para o entendimento do comportamento do sistema. Além disso, os testes em malha fechada, utilizando os controladores propostos pelos autores, validaram plenamente o funcionamento do supervisório em conjunto com o Arduino Uno, evidenciando o comportamento do nível em ambos os casos.

Destaca-se que o segundo teste dos controladores demonstrou um desempenho superior, com um *overshoot* reduzido de nível em relação ao primeiro teste, atrelado ao valor menor de ganho, e uma estabilização mais rápida no *setpoint* desejado, relacionado diretamente ao valor da parte integral

Em resumo, os principais objetivos do trabalho foram alcançados com êxito, incluindo a coleta de dados em tempo real da planta piloto com o Arduino Uno, a implementação do supervisorio no ScadaBR e a validação dos testes em malha aberta e fechada com controladores previamente desenvolvidos.

5. CONSIDERAÇÕES FINAIS

Além disso, para trabalhos futuros seria interessante a implementação da parte derivativa dos controladores, utilizando um filtro de sinal para a parcela derivativa do sistema, de modo a reduzir o efeito da oscilação da medição do sensor ultrassônico na ação derivativa.

Uma sugestão também seria explorar a integração de outros dispositivos e sensores no sistema, como sensores de temperatura, pressão ou fluxo, para expandir a capacidade de monitoramento e controle da planta piloto. Isso poderia fornecer uma visão mais abrangente do processo e permitir a implementação de estratégias de controle mais sofisticadas.

Outra área de pesquisa promissora seria investigar a aplicação de técnicas de machine learning ou inteligência artificial para aprimorar o desempenho do sistema de controle. Isso poderia envolver o desenvolvimento de modelos preditivos do processo ou a utilização de algoritmos de aprendizado de máquina para ajustar dinamicamente os parâmetros do controlador com base nas condições operacionais em tempo real.

REFERÊNCIAS

ALMEIDA, Yan Almeida.; RODRIGUES, Danilo Borges; SOUZA, Davi Leonardo de. Modeling and digital control of the liquid level of a pilot plant, using the firefly algorithm. **Research, Society and Development**, [S. l.], v. 10, n. 9, p. e50510918166, 2021. DOI: 10.33448/rsd-v10i9.18166. Disponível em: <<https://rsdjournal.org/index.php/rsd/article/view/18166>>. Acesso em: 5 mar. 2024.

BEGA, E. A. *et al.* Instrumentação Industrial. **Editora Interciência**, Rio de Janeiro, 2ª ed. 583 p, 2006.

CARDOSO, Heitor Thurler. **Aplicação do SCADABR para monitoramento de sistemas de controle feedback simulado em ambiente Python**. 2022. 87 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Química) - Universidade Federal Fluminense, Niterói, 2022.

CAVALCANTI, Francisco Alves. **Supervisorio/IHM aplicado ao processo de uma coluna de destilação**. 2008. 68 f. Monografia (Especialização) - Curso de Curso de Especialização em Engenharia de Instrumentação, Centro de Tecnologia e Geociências, Universidade

Federal de Pernambuco, Recife, 2008. Disponível em: <http://www.maxpezzin.com.br/aulas/7_EAC_Projetos_Microcontroladores/Francisco_Monografia_PROMINP_I.pdf>. Acesso em: 31 out. 2023.

FERMINO, Fernando. **Estudo comparativo de métodos de sintonia de controladores PID**. 2014. 90 f. Monografia (Especialização) - Curso de Engenharia Elétrica, Escola de Engenharia de São Carlos, Universidade de São Carlos, São Carlos, 2014. Disponível em: <http://www.tcc.sc.usp.br/tce/disponiveis/18/180500/tce-14012015-162330/publico/Fermino_Fernando.pdf>. Acesso em: 06 nov. 2023.

GIMENES, A. H.; PEREIRA, PRA. **Utilização da Plataforma Arduino como Ferramenta de Controle para um Sistema de Nível de Líquidos**. In: **XI Congresso Brasileiro de Engenharia Química em Iniciação Científica**. Unicamp, Campinas, São Paulo. 2015.

KAZALA, R.; STRACZYNSKI, P.; The Most Important Open Technologies for Design of Cost Efficient Automation Systems. **Elsevier**, IFAC-PapersOnLine, v. 52, n. 25, p. 391–396, 2019.

MIRANDA, Thiago da Cunha Rocha Devesa de. **Uso do SCADABR para desenvolvimento de sistemas supervisórios de processos químicos simulados**. 2021. 94f. Trabalho de Conclusão de Curso (Graduação em Engenharia Química) - Universidade Federal Fluminense, Niterói, 2021

PINTO, Regina Carneiro. **A FORMAÇÃO DO ENGENHEIRO QUÍMICO NO CONTEXTO DA INDÚSTRIA 4.0**. 2020. 37 f. TCC (Graduação) - Curso de Engenharia Química, Departamento de Engenharia Química, Universidade Federal de São Carlos, São Carlos, 2020. Disponível em: <https://repositorio.ufscar.br/bitstream/handle/ufscar/13731/598429_TG_Revisado.pdf?sequence=1&isAllowed=y>. Acesso em: 30 out. 2023.

PITTA, João Vitor Tostes; VALENTINI, Virgínia Borges; Souza, Davi Leonardo de. **Modelagem, projeto e análise do desempenho de controladores PID para um sistema de nível de líquido em um reservatório utilizando o algoritmo bioinspirado de Colônia de Vagalumes**. 2023. 14 f. TCC (Graduação) - Curso de Engenharia Química, Departamento de Engenharia Química, Universidade Federal do Triângulo Mineiro, Uberaba, 2023.

SANTOS, Marcos; MANHÃES, Aline Martins.; LIMA, Angélica Rodrigues. **Indústria 4.0: Desafios e oportunidades para o Brasil**. In: **SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO DE SERGIPE**, 10., 2018, São Cristóvão, SE. Anais do X Simpósio de Engenharia de Produção de Sergipe. São Cristóvão, SE, 2018. p. 317 - 329.

SEBORG, Dale E.; EDGAR, Thomas F.; MELLCHAMP, Duncan A. – **Process Dynamics and Control**, Editora. John Wiley & Sons, 2003.

SOARES, Fabrício Magalhães. **APLICAÇÃO PRÁTICA DE MÉTODOS DE SINTONIA DE CONTROLADORES PID EM UMA PLANTA DIDÁTICA INDUSTRIAL**. 2018. 98 f. Monografia (Especialização) - Curso de Engenharia de Controle e Automação,

Universidade Federal de Ouro Preto, Ouro Preto, 2018. Disponível em: https://monografias.ufop.br/bitstream/35400000/1352/1/MONOGRAFIA_AplicaçãoPrática Métodos.pdf. Acesso em: 31 out. 2023.

SWIECH, Maria. **SINTONIA DE CONTROLADORES PID EM COLUNAS DE DESTILAÇÃO ATRAVÉS DE ALGORITMOS GENÉTICOS**. In: 3º Congresso Brasileiro de P&D em Petróleo e Gás, 2005, Salvador, pg. 1-6.