

Capítulo IV – Implementação de um controle de nível PID no TIA Portal

John Everson Rodrigo Mello Jayme ¹

Fabio Oswaldo de Souza Koyano Filho ²

Wagner de Lima Santos ³

Wesley Candido da Silva⁴

Rodolfo Alexandre Hildebrandt⁵

Ana Cristina Fermino Deschamps ⁶

Cassiana Fagundes da Silva ⁷

Denice Lusa ⁸

Eli Carlos Dal Pupo ⁹

Joair Rossetto Schelela Junior ¹⁰

Jorge Ferreira De Sa Junior ¹¹

Suellen Terroso de Mendonça Ferreira Jacoboski ¹²

Thiago Schaedler Uhlmann ¹³

Tuany Kasiorowski Neves ¹⁴

RESUMO

O controle PID (Proporcional, Integral, Derivativo) é um método amplamente utilizado em engenharia de controle para estabilizar e controlar sistemas dinâmicos. É uma técnica que combina três componentes distintos para ajustar o desempenho do sistema em relação a um valor desejado ou setpoint, para ilustrar a utilização de um controle PID na indústria, é empregado o software TIA Portal da Siemens, uma plataforma de controle para múltiplos CLPs. O CLP selecionado para esse fim é o SIMATIC S7-1200 (1214C DC/DC/DC), com uma programação em ladder. Dentro dessa programação, são aplicados métodos de tratamento de variáveis e a implementação do controle PID é realizada por meio de blocos de programação específicos. Uma IHM (KTP600 Basic color PN) é utilizada para possibilitar o acionamento do inversor de controle de velocidade da bomba, permitindo a seleção do setpoint desejado e a visualização do volume do tanque. Além disso, é fornecido um gráfico em tempo real que exibe as variáveis de valor de referência, amplitude de acionamento do motor e a variável controlada.

Palavras-chave: Controle PID, Software TIA Portal, Controle de Nível.

¹ Estudante Engenharia Elétrica UniSENAI PR -Campus Londrina, jonhermj@hotmail.com

² Estudante Engenharia Elétrica UniSENAI PR -Campus Londrina, fabiofoyano@outlook.com

³ Estudante Engenharia Elétrica UniSENAI PR -Campus Londrina, delimasantoswagner@gmail.com

⁴ Docente UniSenaiPR - Campus Londrina, wesley.candido@sistemafiep.org.br

⁵ Docente UniSenaiPR - Campus Londrina, rodolfo.hildebrandt@sistemafiep.org.br

⁶ Docente UniSenaiPR - Campus da Indústria, ana.deschamps@sistemafiep.org.br

⁷ Docente UniSenaiPR - Campus São José dos Pinhais, cassiana.silva@sistemafiep.org.br

⁸ Docente UniSenaiPR - Campus São José dos Pinhais, denice.lusa@sistemafiep.org.br

⁹ Docente UniSenaiPR - Campus São José dos Pinhais, eli.pupo@sistemafiep.org.br

¹⁰ Docente UniSenaiPR - Campus São José dos Pinhais, joair.junior@sistemafiep.org.br

¹¹ Docente UniSenaiPR - Campus CIC, jorge.junior@sistemafiep.org.br

¹² Docente UniSenaiPR - Campus CIC, suellen.jacoboski@sistemafiep.org.br

¹³ Docente UniSenaiPR - Campus São José dos Pinhais, thiago.uhlmann@sistemafiep.org.br

¹⁴ Docente UniSenaiPR - Campus São José dos Pinhais, tuany.neves@sistemafiep.org.br

INTRODUÇÃO

O controle PID pode ser implementado de diversas maneiras e métodos, abrangendo tanto a implementação digital por meio de microcontroladores quanto a implementação analógica com o uso de componentes eletrônicos específicos.

Na implementação digital, utiliza-se um microcontrolador para realizar os cálculos e as operações necessárias para o controle PID. O microcontrolador lê o valor medido, calcula o erro em relação ao valor desejado e, em seguida, determina a saída de controle adequada com base nos parâmetros do PID (K_p , K_i , K_d) e nas contribuições proporcional, integral e derivativa. Essa abordagem é amplamente utilizada devido à flexibilidade e facilidade de ajuste dos parâmetros do PID.

Na implementação analógica, o controle PID é realizado através de componentes eletrônicos dedicados. Componentes como amplificadores operacionais, circuitos integradores e circuitos derivativo são utilizados para implementar as contribuições proporcional, integral e derivativa. Essa abordagem é adequada para sistemas que requerem respostas em tempo real e podem ser vantajosos em aplicações específicas.

Além das implementações digital e analógica, também existem outras formas de implementar o controle PID. Isso inclui a utilização de controladores programáveis dedicados, que executam as operações do controle PID e oferecem recursos avançados de controle. Também é possível implementar o controle PID em software, onde os cálculos e operações são realizados por meio de algoritmos em um computador. Essa abordagem é comum em simulações e sistemas de controle baseados em computador.

A escolha da forma de implementação do controle PID depende das características do sistema, dos requisitos de desempenho, da flexibilidade necessária e de outros fatores específicos da aplicação. Cada método possui suas próprias vantagens e desafios, e é importante selecionar a abordagem mais adequada para atender às necessidades do sistema em questão.

O software TIA Portal, da Siemens, possui um bloco específico de controle PID que é projetado para facilitar a implementação do controle PID. Esse bloco contém cálculos internos e funcionalidades para controle de variáveis analógicas ou digitais. O bloco de controle PID no TIA Portal permite ajustar os parâmetros do PID, como os coeficientes de ganho proporcional, integral e derivativo, diretamente no software. Essa configuração pode ser feita em valores absolutos ou em porcentagem, dependendo das necessidades do sistema.

Além disso, o TIA Portal oferece modelos de IHM (Interface Homem-Máquina) para controle e visualização de processos em tempo real. Essas IHMs fornecem uma interface gráfica intuitiva que permite aos operadores monitorar e interagir com o sistema de controle PID. Eles podem ajustar

os valores de referência, visualizar as variáveis do processo em tempo real e realizar outras operações de controle conforme necessário.

O uso do TIA Portal simplifica o desenvolvimento e a implementação do controle PID, fornecendo ferramentas e recursos específicos para essa finalidade. Isso permite um ajuste mais eficiente do PID e uma interface de controle amigável para os operadores do sistema.

5 MATERIAIS E MÉTODOS

O TIA Portal, desenvolvido pela Siemens, é um software de engenharia que oferece suporte para a implementação do controle PID em sistemas industriais. O TIA Portal possui recursos específicos para configurar e ajustar controladores PID de forma eficiente.

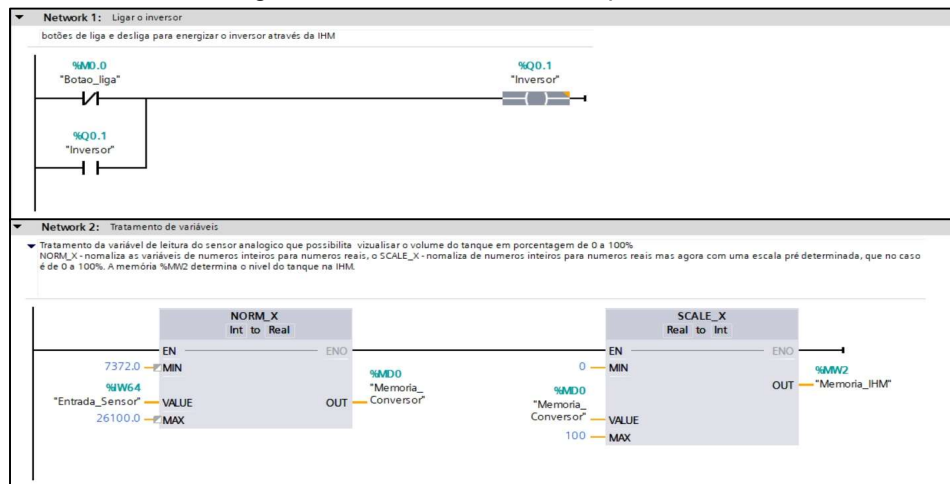
Para utilizar o controle PID no TIA Portal, é necessário acessar a seção de programação do software. Dentro dessa seção, é possível encontrar blocos de função dedicados ao controle PID. Esses blocos de função possuem parâmetros configuráveis dos ganhos proporcional, integral e ganho derivativo. É possível ajustar os ganhos de forma manual ou utilizar recursos automáticos de sintonia.

Para executar o bloco de controle PID de forma periódica e regular, é necessário adicionar uma função de interrupção cíclica (*cyclic interrupt*) no programa. Essa função permite que o bloco de controle PID seja executado em intervalos regulares e predefinidos.

A função de interrupção cíclica é responsável por determinar a frequência de atualização do controle PID. Ela define o período de tempo no qual o bloco de controle PID será executado e realizará os cálculos necessários para o controle do sistema. Ao adicionar a função de interrupção cíclica no programa, é possível configurar o período desejado para a execução do bloco de controle PID. Esse período pode variar de acordo com as características do sistema e os requisitos de tempo de resposta.

Uma vez configurada a função de interrupção cíclica, o bloco de controle PID será acionado regularmente dentro desse período definido. Isso permite que o controle PID monitore e ajuste o sistema de forma contínua, proporcionando um controle mais preciso e eficiente. A adição da função de interrupção cíclica no programa do TIA Portal garante que o bloco de controle PID seja executado de forma regular e periódica, garantindo o controle adequado do sistema em tempo real.

Figura 1 – Bloco Main Utilizado para Controle de Nível



Fonte: dos Autores.

Dentro do bloco principal (*Main*), foi adicionado um conjunto de botões que permite a ligação e desligamento do inversor responsável pelo controle da velocidade do fluxo de água por meio da bomba. Para implementar essa funcionalidade, é necessário utilizar uma entrada digital para cada botão. A entrada digital é responsável por capturar o estado (ligado/desligado) dos botões na IHM. Quando o botão de ligar é pressionado, a entrada digital correspondente é acionada, enviando um sinal para o programa de controle. Esse sinal é utilizado para ativar o inversor, permitindo que ele comece a controlar a velocidade do fluxo de água através da bomba.

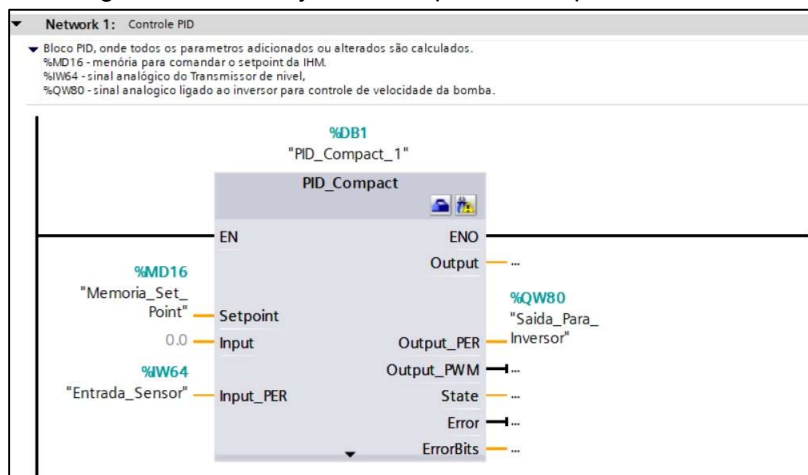
Da mesma forma, ao pressionar o botão de desligar, a entrada digital associada é acionada, enviando um sinal para o programa de controle. Esse sinal é utilizado para desativar o inversor, interrompendo o controle da velocidade do fluxo de água. Essa funcionalidade dos botões de ligar e desligar permite aos operadores iniciar ou parar o funcionamento do inversor, proporcionando controle manual sobre a velocidade do fluxo de água através da bomba. Isso pode ser utilizado para ajustar a vazão de água de acordo com as necessidades do processo ou para desligar a bomba quando não é mais necessário o bombeamento de água.

Para possibilitar a visualização do volume do tanque em porcentagem de 0 a 100%, é necessário realizar o tratamento da variável de leitura do sensor analógico. Isso pode ser feito através de duas etapas principais: normalização da variável e atribuição do valor normalizado à memória correspondente na IHM. A primeira etapa envolve a normalização da variável de leitura do sensor. Existem duas abordagens comumente utilizadas: a normalização geral, representada pela função *NORM_X*, e a normalização com escala específica, representada pela função *SCALE_X*. A função *NORM_X* é responsável por transformar as variáveis de números inteiros em números reais. Ela mapeia a faixa de valores da variável de leitura do sensor para o intervalo entre 0 e 1.

A função SCALE_X, por sua vez, normaliza as variáveis de números inteiros para números reais, mas com uma escala pré-determinada. Nesse caso específico, a escala é definida como 0 a 100%. Assim, a função SCALE_X mapeia a faixa de valores da variável de leitura do sensor para o intervalo entre 0 e 100%. Uma vez que a variável de leitura do sensor é normalizada, o próximo passo é atribuir o valor normalizado à memória correspondente na IHM. No caso mencionado, a memória %MW2 é utilizada para armazenar o nível do tanque em porcentagem. Portanto, o valor normalizado obtido após a etapa de normalização é atribuído à memória %MW2, permitindo a visualização do volume do tanque em porcentagem na IHM.

Essas etapas de tratamento da variável e atribuição do valor normalizado à memória na IHM garantem que o volume do tanque seja corretamente representado em uma escala de 0 a 100%, facilitando a sua visualização e monitoramento.

Figura 2 – Bloco Cyclic Interrupt Utilizado para Controle de Nível

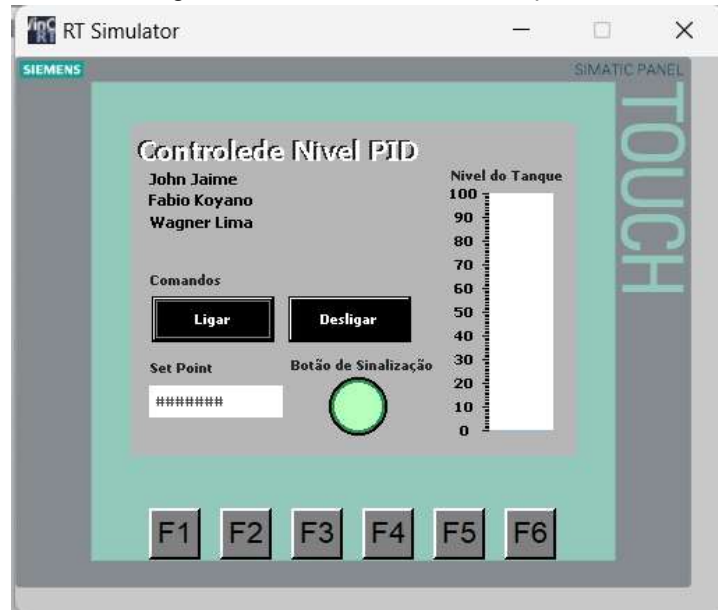


Controle PID - PLC_1 [CPU 1214C DC/DC] - PLC tags									
Tags									
PLC tags									
	Name	Tag table	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	Inversor	Default tag table	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Liga o Inversor
2	Botao_liga	Default tag table	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Botão na IHM que Liga o Inversor
3	Botao_Desliga	Default tag table	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Botão na IHM que Desliga o Inversor
4	Entrada_Sensor	Default tag table	Int	%IW64		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sinal Analógico Vindo do Transmissor
5	Saida_Para_Inversor	Default tag table	Word	%QW80		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sinal Analógico de Saída do Clp Para o Inversor
6	Memoria_Set_Point	Default tag table	Real	%MD16		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Memória que Habilita o Setpoint Através da IHM
7	Memoria_Conversor	Default tag table	Real	%MD0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Memória que Normaliza o Sinal Analógico
8	Memoria_IHM	Default tag table	Int	%MW2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Memória que Insere no Tanque da IHM o Valor Normalizado
9	<Add new>					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Fontes: dos autores

A imagem acima representa a lista dos endereços utilizados na programação em ladder do CLP. Cada endereço desse tem um papel importante dentro da lógica, variáveis que foram ligadas na IHM para retornar um feedback para o operador do sistema.

Figura 3 – Interface Homem Máquina



Fonte: dos autores

Na imagem 4 podemos ver a imagem da IHM do processo. Com fácil visualização do nível do tanque, botões de controle de início e fim do processo e um botão de sinalização. Podemos ver também a área dedicada a mudança do setpoint de processo. Para essa visualização foi necessário realizar conexões internas com o código escrito utilizando alguns dos endereços mencionados anteriormente na imagem 3.

Figura 4 – IHM em Funcionamento



Fonte: dos autores

Na imagem acima é possível observar a IHM em funcionamento na bancada didática utilizada para controle do processo. A Figura 5 ilustra a bancada utilizada para a implementação do controle clássico PI.

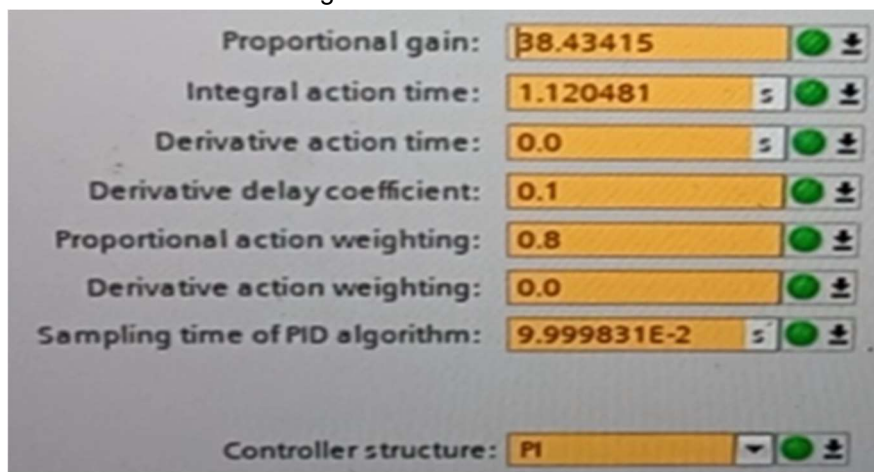
Figura 5 – Bancada Utilizada para Aplicação do Controle PI



Fonte: dos autores

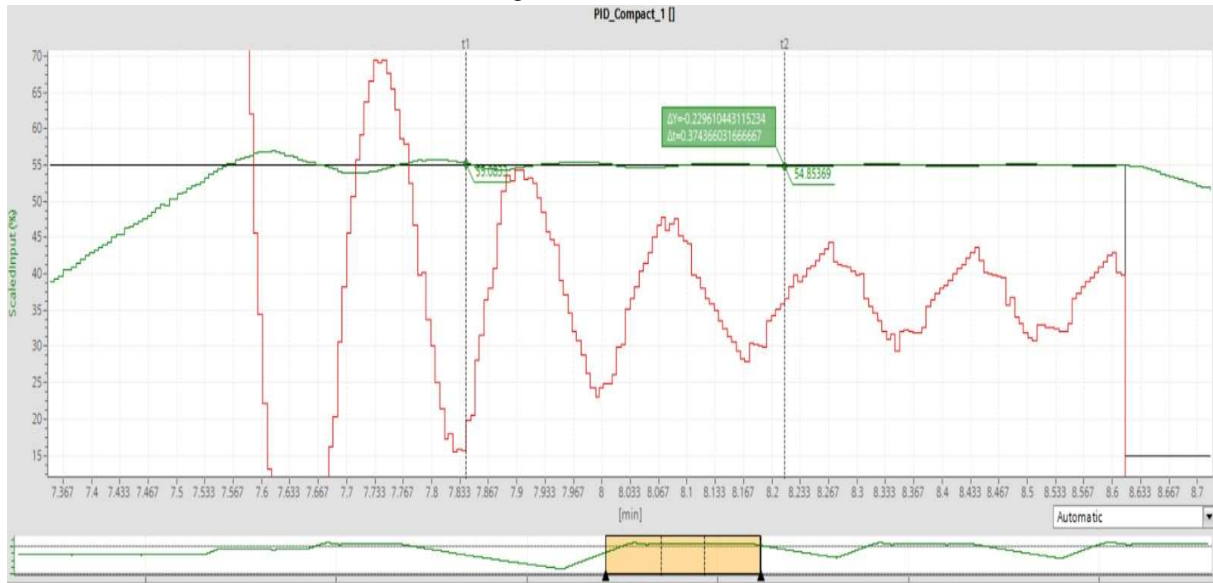
Utilizando a bancada didática conseguimos visualizar de forma prática as alterações nas saídas e entradas de acordo com os ganhos aplicados no sistema Proporcional, Integral e Derivativo, como observado nas Figuras 6 e 7.

Figura 6 – Parâmetros PI



Fonte: dos autores

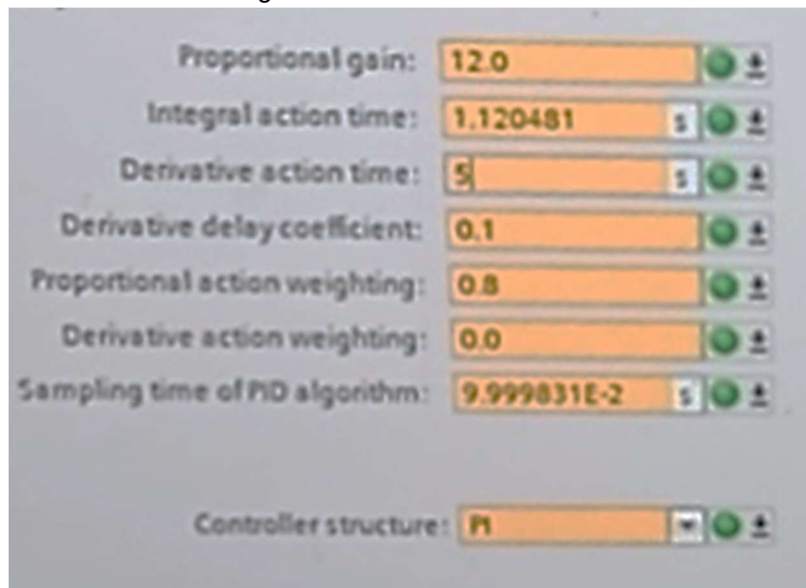
Figura 7 – Gráfico PI



Fonte: dos autores

Pode-se observar na Figura 7, uma oscilação que representa o acionamento da bomba da bancada, essa variação deu-se devido o ganho derivativo estar em zero, parâmetros escolhidos. Assim, foi implementado um controle do tipo PID, conforme ilustra a Figura 8.

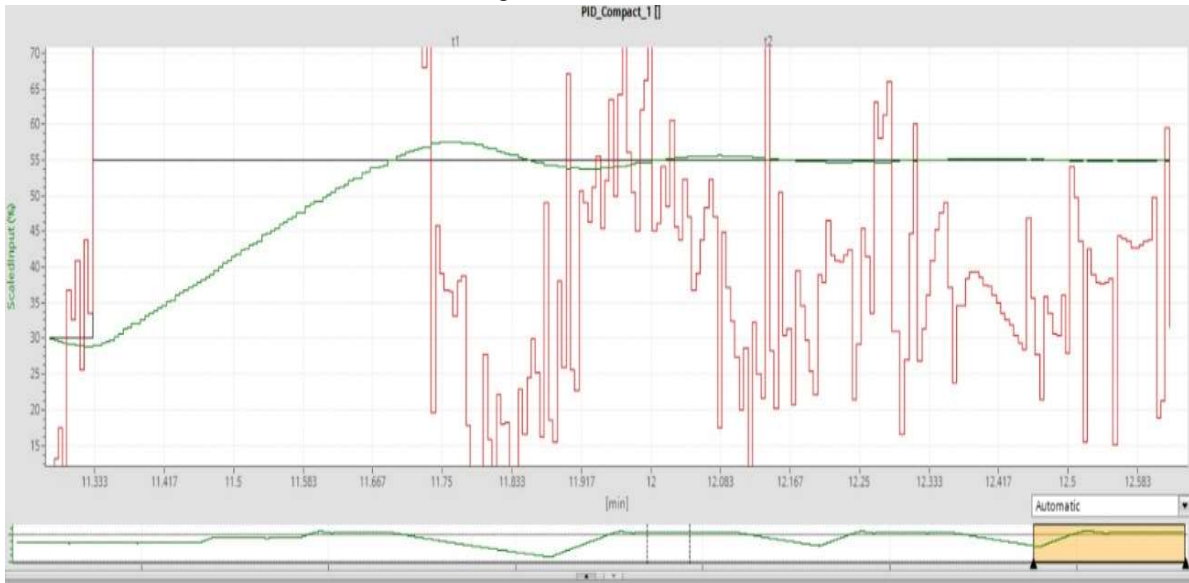
Figura 8 – Parâmetros do PID



Fonte: dos autores

Com a alteração do parâmetro derivativo de zero para cinco, podemos notar a diferença no acionamento da bomba conforme imagem abaixo.

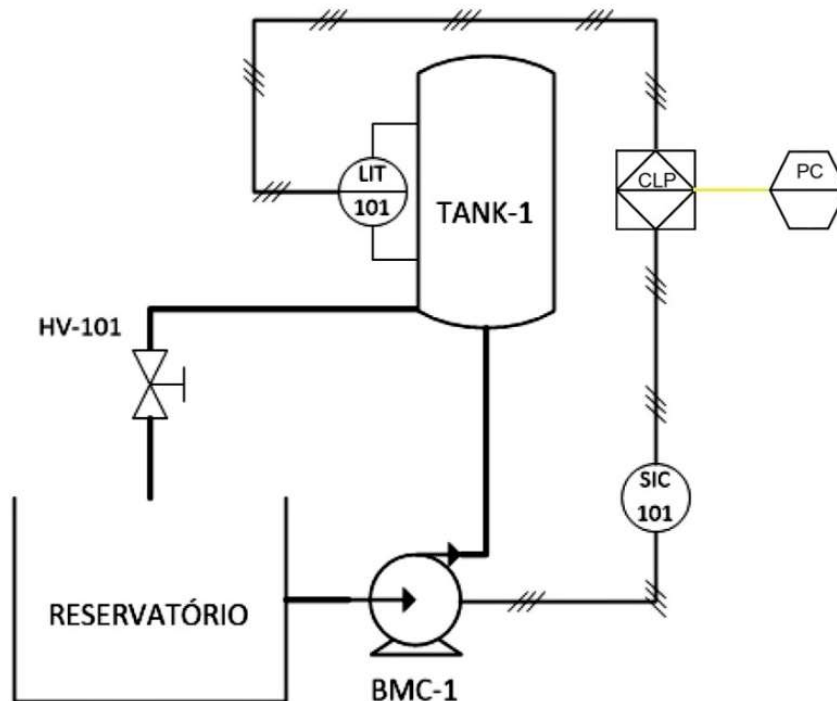
Figura 9 – Saída PID



Fonte: dos autores

Ao realizarmos uma breve comparação entre os sistemas PI e PID, podemos notar um comportamento totalmente diferente no acionamento da bomba da bancada. Com o ganho derivativo acionada o processo de controle fica mais preciso, alcançando o setpoint de uma forma melhor e controlando mais precisamente a saída para o processo.

Figura 10 – Diagrama do Controle de Nível do Reservatório



Fonte: dos autores

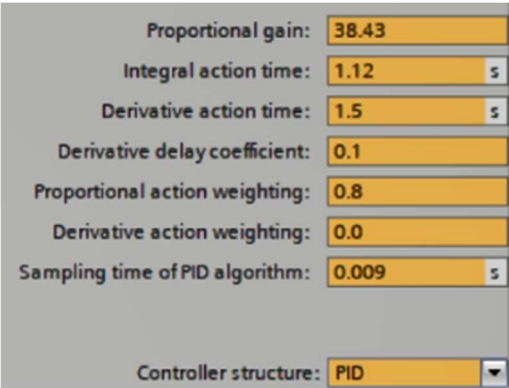

A ilustração acima retrata o diagrama P&ID da planta didática que será controlada. É possível observar os componentes de controle, como o Controlador Lógico Programável (CLP) e o

sensor, bem como os elementos controlados, como a bomba e o nível do reservatório. O HV-101 é uma válvula manual que pode ser acionada caso seja necessário intervir de alguma forma, independentemente da natureza da intervenção. O componente SIC-101 representa o inversor de frequência, controlado pelo CLP, que regula a bomba proporcionalmente ao nível do reservatório.

2. RESULTADOS

Quando aplicamos um ganho derivativo para esse processo, como apresentado anteriormente, possuímos um acionamento da bomba de controle mais brusco, e assim, prejudicando a longo prazo os componentes físicos do processo. Utilizando um ganho derivativo de 1.5, notamos um OverShooting bem como, o acionamento robusto da bomba:

Tabela 1 – Ajuste do Ganho Derivativo

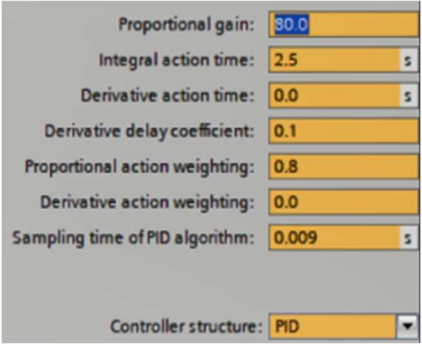
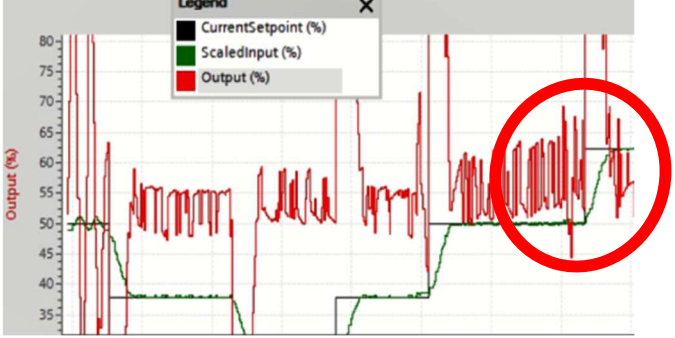
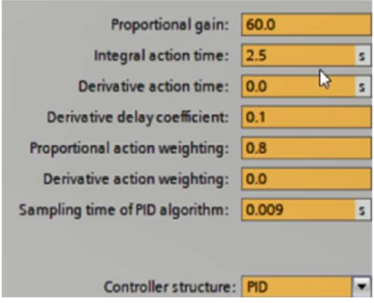
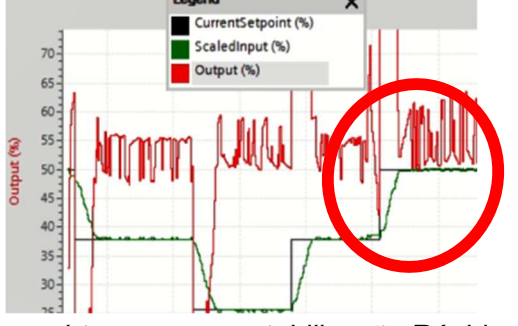
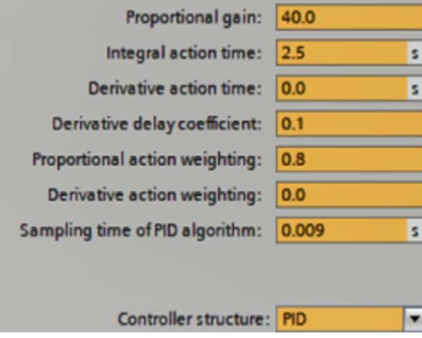
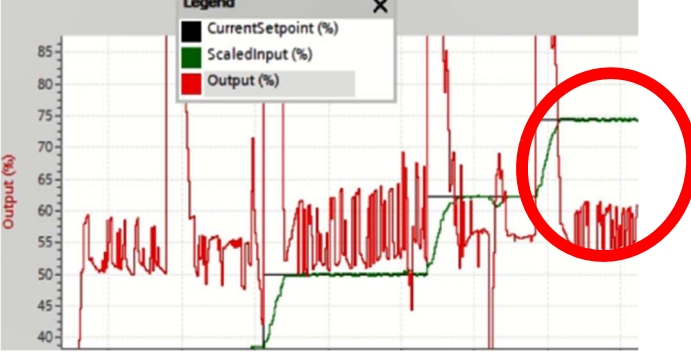
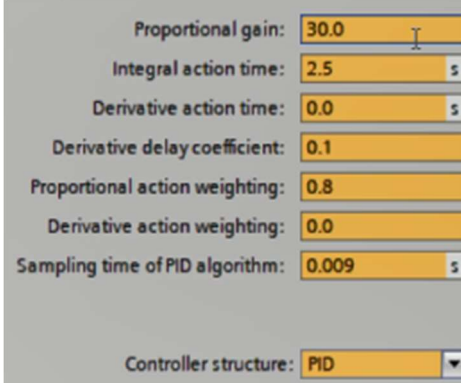
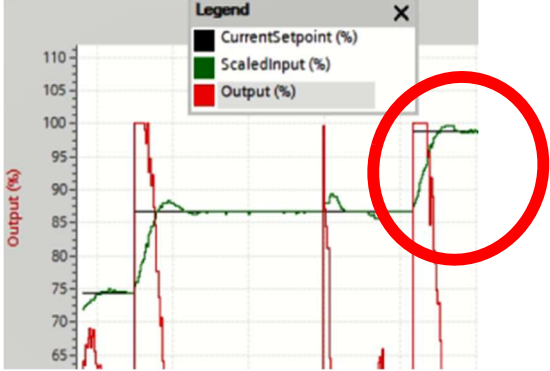
Parâmetros PI	Comportamento com redução do SetPoint
 <p>*Comprovando que o ganho derivativo não é recomendado para essa aplicação.</p>	 <p>*Notamos um OverShooting consideravelmente alto.</p>

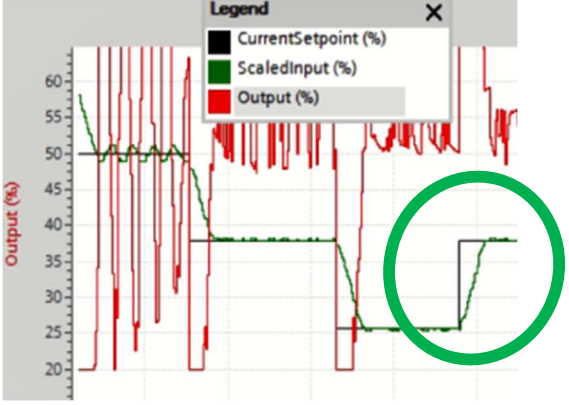
Fonte: dos autores

Sendo assim, seguimos sem o ganho derivativo nos ajustes dos ganhos integrativos e proporcional, devido ao método de controle onde buscamos um controle de nível, recomenda-se apenas o uso de um sistema Proporcional Integrativo por ser um sistema de 1 ordem.

Foi escolhido como primeiro ganho a ser regulado, o ganho proporcional, onde, após vários ajustes foi possível obter o melhor valor olhando para o overshooting e para o tempo de normalização do nível conforme Setpoint.

Tabela 2 – Ajuste do Ganho Proporcional

Parâmetros PI	Comportamento com aumento no SetPoint
 <p>Proportional gain: 30.0 Integral action time: 2.5 s Derivative action time: 0.0 s Derivative delay coefficient: 0.1 Proportional action weighting: 0.8 Derivative action weighting: 0.0 Sampling time of PID algorithm: 0.009 s Controller structure: PID</p>	 <p>*Observamos uma estabilização Rápida sem OverShooting.</p>
 <p>Proportional gain: 60.0 Integral action time: 2.5 s Derivative action time: 0.0 s Derivative delay coefficient: 0.1 Proportional action weighting: 0.8 Derivative action weighting: 0.0 Sampling time of PID algorithm: 0.009 s Controller structure: PID</p>	 <p>*Tambem obtemos uma estabilização Rápida sem OverShooting.</p>
 <p>Proportional gain: 40.0 Integral action time: 2.5 s Derivative action time: 0.0 s Derivative delay coefficient: 0.1 Proportional action weighting: 0.8 Derivative action weighting: 0.0 Sampling time of PID algorithm: 0.009 s Controller structure: PID</p>	 <p>*Continuamos com uma estabilização Rápida sem OverShooting.</p>
 <p>Proportional gain: 30.0 Integral action time: 2.5 s Derivative action time: 0.0 s Derivative delay coefficient: 0.1 Proportional action weighting: 0.8 Derivative action weighting: 0.0 Sampling time of PID algorithm: 0.009 s Controller structure: PID</p>	 <p>*Observamos uma estabilização Rápida com OverShooting.</p>

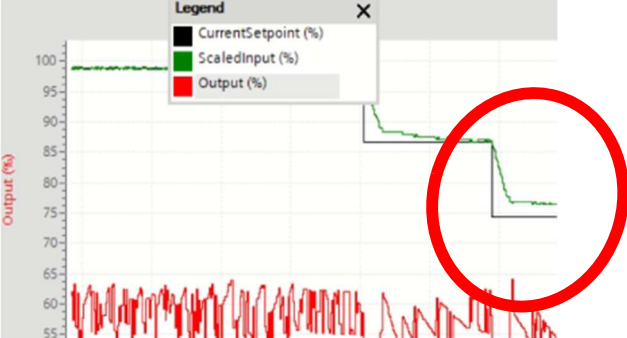
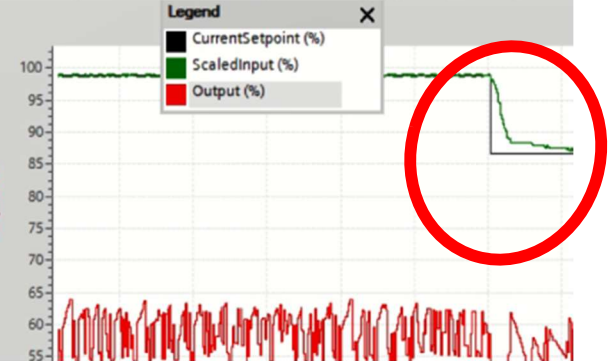
<p>Proportional gain: <input style="width: 100%;" type="text" value="38.43"/></p> <p>Integral action time: <input style="width: 100%;" type="text" value="2.5"/> s</p> <p>Derivative action time: <input style="width: 100%;" type="text" value="0.0"/> s</p> <p>Derivative delay coefficient: <input style="width: 100%;" type="text" value="0.1"/></p> <p>Proportional action weighting: <input style="width: 100%;" type="text" value="0.8"/></p> <p>Derivative action weighting: <input style="width: 100%;" type="text" value="0.0"/></p> <p>Sampling time of PID algorithm: <input style="width: 100%;" type="text" value="0.009"/> s</p> <p>Controller structure: <input style="width: 100%;" type="text" value="PID"/></p>	 <p>*Observamos uma estabilização Rápida sem OverShooting.</p>
--	--

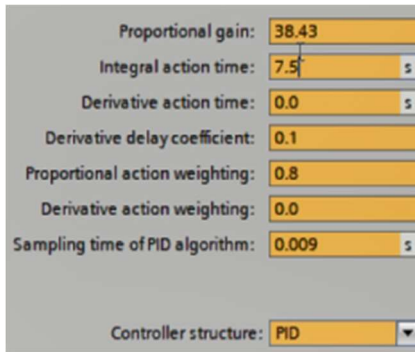
*Notamos que o valor próximo de 40 no ganho proporcional é o mais adequado. Sendo assim, padronizado em 38.43.

Fonte: dos autores

Uma vez encontrado o melhor ganho para o ganho proporcional, visando uma rápida normalização entre nível do tanque e Setpoint seguimos para as alterações dos ganhos integrativos afins de se obter a melhor performance para o processo.

Tabela 3 – Ajuste do Ganho Proporcional

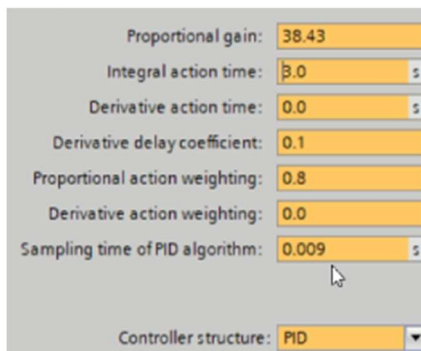
Parâmetros PI	Comportamento com redução do SetPoint
<p>Proportional gain: <input style="width: 100%;" type="text" value="38.43"/></p> <p>Integral action time: <input style="width: 100%;" type="text" value="38.0"/> s</p> <p>Derivative action time: <input style="width: 100%;" type="text" value="0.0"/> s</p> <p>Derivative delay coefficient: <input style="width: 100%;" type="text" value="0.1"/></p> <p>Proportional action weighting: <input style="width: 100%;" type="text" value="0.8"/></p> <p>Derivative action weighting: <input style="width: 100%;" type="text" value="0.0"/></p> <p>Sampling time of PID algorithm: <input style="width: 100%;" type="text" value="0.009"/> s</p> <p>Controller structure: <input style="width: 100%;" type="text" value="PID"/></p>	 <p>*Observamos uma estabilização lenta, sem OverShooting.</p>
<p>Proportional gain: <input style="width: 100%;" type="text" value="38.43"/></p> <p>Integral action time: <input style="width: 100%;" type="text" value="15.0"/> s</p> <p>Derivative action time: <input style="width: 100%;" type="text" value="0.0"/> s</p> <p>Derivative delay coefficient: <input style="width: 100%;" type="text" value="0.1"/></p> <p>Proportional action weighting: <input style="width: 100%;" type="text" value="0.8"/></p> <p>Derivative action weighting: <input style="width: 100%;" type="text" value="0.0"/></p> <p>Sampling time of PID algorithm: <input style="width: 100%;" type="text" value="0.009"/> s</p> <p>Controller structure: <input style="width: 100%;" type="text" value="PID"/></p>	 <p>*Observamos uma estabilização mais rápida, sem OverShooting.</p>



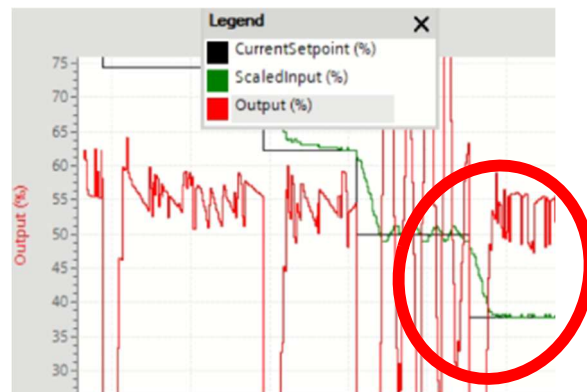
*Reduzindo 7,5 no ganho integrativo.



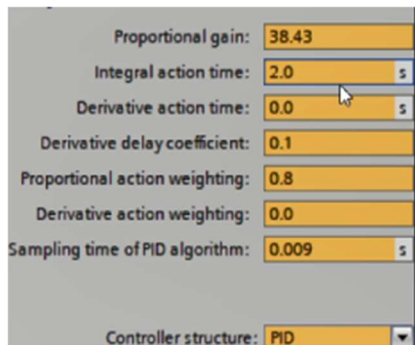
*Observamos uma estabilização mais rápida, sem OverShooting. Contudo, oscilações no sinal.



*Reduzindo 4,5 no ganho integrativo.



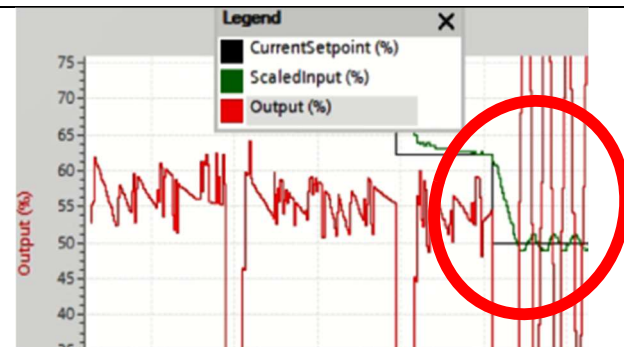
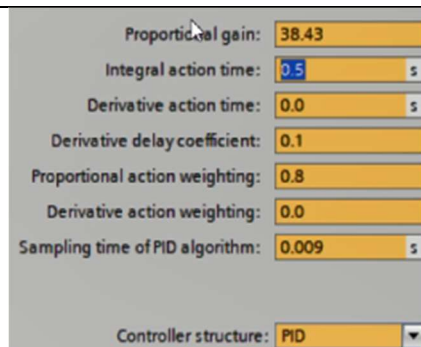
*Observamos uma estabilização mais rápida, sem OverShooting. Contudo, oscilações no sinal.

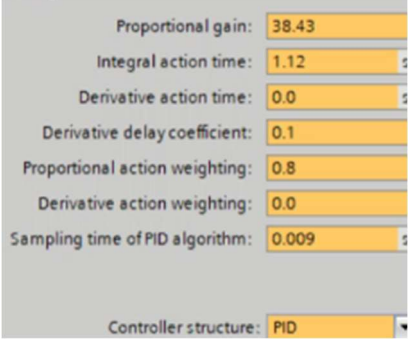
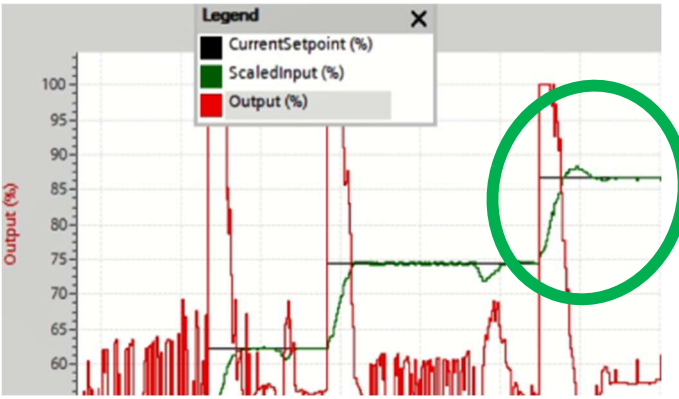


*Reduzindo 2,5 no ganho integrativo.



*Observamos uma estabilização mais rápida, sem OverShooting. Contudo, oscilações no sinal.



<p>*Reduzindo 1,5 no ganho integrativo.</p>	<p>*Observamos uma estabilização mais rápida, sem OverShooting. Contudo, muitas oscilações no sinal.</p>
 <p>*Diminuindo o ganho integrativo mantendo o melhor parâmetro no ganho proporcional.</p>	 <p>*Observamos uma estabilização Rápida com grande OverShooting.</p>

Fonte: dos autores

E assim, após todos os testes serem concluídos, conseguimos encontrar os parâmetros em:

- ✓ Ganho proporcional: 38,43;
- ✓ Ganho integrativo: 1.12;
- ✓ Ganho derivativo: 0,0;

3 CONCLUSÕES

Em conclusão, este artigo explora a comparação entre os sistemas de controle PI (Proporcional-Integral) e PID (Proporcional-Integral-Derivativo) em relação à aplicação de controle de vazão em bombas. Embora ambos os sistemas sejam amplamente utilizados na indústria para garantir a estabilidade e precisão do processo, ficou evidente que o controle PI é mais adequado para essa aplicação específica.

Ao analisar os resultados e considerar os principais aspectos, como tempo de resposta, estabilidade e capacidade de controle da vazão, o controle PI se mostrou uma escolha mais eficiente. O controle PID, ao adicionar o termo derivativo, introduz complexidade adicional, tornando a sintonia mais desafiadora e aumentando o risco de oscilações indesejadas e instabilidades.

Por outro lado, o controle PI testado será capaz de fornecer um bom desempenho de controle, mantendo a estabilidade e evitando oscilações excessivas. Além disso, a simplicidade do controle PI facilita a sintonia e o ajuste, o que pode ser particularmente vantajoso em sistemas com variáveis de processo menos dinâmicas, como o controle de vazão em bombas.

É importante destacar que a escolha entre o controle PI e PID depende das características específicas do sistema e das necessidades de controle. Em alguns casos, quando o processo é mais complexo e é necessário ter uma resposta em tempo real, o controle PID pode ser mais adequado. Por isso, é essencial analisar cuidadosamente as características e requisitos do sistema antes de decidir qual tipo de controle usar.

Embora o controle PID seja amplamente utilizado em muitas aplicações de controle de processos, quando se trata de controlar o fluxo de bombas, o controle PI eficiente é uma opção melhor. Ele oferece estabilidade e precisão sem a complexidade adicional do termo derivativo.

REFERÊNCIAS

Sampaio, F., Cuenca, M., Rodrigues, R., & Silva, W. C. (2022). Metodologia para simulação da Lógica Fuzzy através do SCILAB SciFLTEditor. SENAI Londrina PR, 1(1), pp. 1-18.

Autodesk. (2023). Tinkercad [Simulação de Arduino, Módulo: eletrônica]. Recuperado de <https://www.tinkercad.com/>

TEDESCO, Patricia; GERMANO, Vasconcelos. Lógica Fuzzy. Recuperado em 24 de abril de 2023, de <https://www.cin.ufpe.br/~if684/EC/aulas/Aula-logica-Fuzzy-SI.pdf>.

ALVES, Otávio. eFLL - Fuzzy Logic Library for Arduino. Disponível em: <https://github.com/alvesoj/eFLL>. Acesso em: 24 de abril de 2023.

ARDUINO. Software. Disponível em: <https://www.arduino.cc/en/software>. Acesso em: 27 de abril de 2023.

ARDUINO. Tutorials. Disponível em: <https://www.arduino.cc/en/Tutorial/HomePage>. Acesso em: 27 de abril de 2023.