10.18605/2175-7275/cereus.v16n3p81-97

<< Recebido em: 10/04/2024 Aceito em: 18/08/2024. >>

ARTIGO ORIGIINAL

REVISTA CEREUS
ISSN: 2175-7275

Modelagem, projeto e análise do desempenho de controladores PID para um sistema de nível de líquido em um reservatório utilizando o algoritmo bio-inspirado de Colônia de Vagalumes

Modeling, design and performance analysis of PID controllers for a liquid level system in a reservoir using the bio-inspired algorithm of Firefly Colony

João Vitor Tostes Pitta[1], Virgínia Borges Valentini[2], Davi Leonardo de Souza [3].

[1] Aluno do Curso de Graduação em Engenharia Química, Universidade Federal do Triângulo Mineiro, Uberaba, Minas Gerais, Brasil. Orcid: https://orcid.org/0000-0002-2375-7984

E-mail: jtostespitta59@gmail.com

[2]Aluna do Curso de Graduação em Engenharia Química, Universidade Federal do Triângulo Mineiro, Uberaba, Minas Gerais, Brasil. Orcid: https://orcid.org/0009-0001-9989-6112

E-mail: virginiabvalentini@gmail.com

[3]Professor, Universidade Federal do Triângulo Mineiro, Uberaba, Minas Gerais, Brasil. Orcid: https://orcid.org/0000-0002-1995-9057

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

RESUMO

O objetivo principal do trabalho é encontrar o melhor modelo através da função de transferência que prediz os dados coletados por meio de uma aquisição de dados em *Python* e, através do algoritmo bio-inspirado Colônia de Vagalumes, obter o melhor controlador PI e PID para o sistema. Foi realizada a implementação dos controladores através do computador e microcontrolador (Arduino UNO), para testar e validar o projeto. O controlador PI foi mais eficiente e mais rápido em controlar o sistema no *setpoint* em relação ao PID pelo fato de ter apresentado oscilações menores.

Palavras-chave: Controladores. Função de transferência. *Python*. Curva de reação.

ABSTRACT

The main objective of this work is to find the best model through the transfer function that predicts the data collected through a data acquisition in Python and, through the bioinspired algorithm Firefly Colony, to obtain the best PI and PID controller for the system. The implementation of the controllers was done through the computer and microcontroller (Arduino UNO), to test and validate the project. The PI controller was more efficient and faster in controlling the system at the setpoint in relation to the PID because it presented smaller oscillations.

Keywords: Controllers. Transfer function. Python. Reaction curve.

1. INTRODUÇÃO

É devido à extrema necessidade de se trabalhar com problemas mais realísticos e que exigem mais precisão e exatidão, que nos últimos anos projetos de sistemas de engenharia vem se preocupando mais em incorporar a utilização de técnicas de otimização computacionais. Segundo Cologni (2008), uma preocupação crescente vem tomando espaço das ações globais: a redução da emissão de poluentes, visando à desaceleração do ritmo atual de aquecimento global.

Com isso, almeja-se que os controladores industriais com sintonia automática e avaliação de desempenho possam contribuir na busca por melhores resultados tanto do ponto de vista econômico quanto ecológico e ambiental. De acordo com Ogata (2010), muitos sistemas de controle industrial das décadas de 1940 e 1950 usavam controladores PID no controle de variáveis, como pressão, temperatura e vazão.

Ao passo que os sistemas modernos se tornaram mais complexos (com muitas entradas e saídas), a teoria clássica de controle, ou seja, que trata sistemas com uma entrada e uma saída, tornou-se insuficiente. Então, de acordo com Nogueira (2023), a partir de 1960 a teoria de controle moderno foi desenvolvida para atender a complexidade destes sistemas simplificando o projeto de controle, uma vez que se baseia no modelo de um sistema de controle real. Já na década de 1980, com a disponibilidade de computadores digitais, os métodos heurísticos, estocásticos e populacionais ganharam força no meio industrial e acadêmico.

Neste contexto, as técnicas de inteligência computacional inspiradas na natureza, tem registrado várias aplicações em áreas distintas da ciência e engenharia para a resolução do problema de otimização (LOBATO, 2008). As principais características dessas técnicas são: concepção conceitual simples, facilidade de implementação, a capacidade de escapar de ótimos locais, são facilmente acoplados a outros softwares e ao fato de não precisarem de informação sobre o gradiente das funções para a determinação do ponto de mínimo.

De acordo com Oliveira MDM et al. (2020), os métodos de resolução de problemas de otimização podem ser classificados em dois tipos: determinísticos (ou clássicos) e não-determinísticos (meta-heurísticos, evolutivos ou randômicos). Os métodos clássicos possuem uma convergência prematura devido à dificuldade que possuem de lidar com mínimos locais. Enquanto os métodos não determinísticos são métodos evolutivos, com estratégias de solução do problema baseada em comportamentos observados na natureza, ou bio-inspirados.

Segundo Duarte (2015), os algoritmos populacionais funcionam da seguinte forma: a partir de um conjunto de soluções iniciais são realizadas operações entre elas até que seja encontrada uma solução ótima para o problema. Dentre estes algoritmos, grande parte foram propostos a partir de fenômenos da natureza e por isso, são considerados bio-inspirados, em que possuem como característica principal encontrar várias soluções ótimas em uma única simulação, devido a sua abordagem de pesquisa baseada na população.

Um algoritmo bastante conhecido e utilizado no presente trabalho é o Algoritmo Colônia de Vagalumes (ACV) que, de acordo com Pereira et al. (2020), caracteriza-se por ser um método de otimização global com aplicações em diversos campos das ciências e seu desempenho é fortemente afetado pelos parâmetros de entrada do algoritmo.

De acordo com Swiech, et al 2004, para manter variáveis dentro dos limites desejáveis são necessários controladores proporcionais (P), integrais (I) e derivativos (D), no qual a ação proporcional atua diretamente no ganho do sistema, a ação integral está relacionada à melhor precisão de resposta, ou seja, baseia-se no tempo em que o erro acontece, e a ação derivativa ajusta a variável apoiando-se na taxa de variação do erro. Por isso, a seleção do controlador deve depender das condições operativas do sistema e de especificações de performance (LOURENÇO, 1997).

Ainda, é importante ressaltar que estas ações de controle não são utilizadas de forma isolada. A união destes três modos básicos de controle contínuo produz um dos mais eficientes algoritmos de controle já desenvolvidos, o controlador PID, pois ele concilia simplicidade e atendimento às necessidades de controle (FACCIN, 2004).

Diante do exposto, o trabalho tem como objetivo geral, encontrar o melhor modelo através da função de transferência que prediz os dados coletados por meio de uma aquisição de dados em Python e, através do algoritmo bio-inspirado Colônia de Vagalumes, projetar o melhor controlador PI e PID para o sistema. O objetivo específico, por sua vez, analisar e comparar o desempenho dos controladores para o sistema em questão.

2. MATERIAIS E MÉTODOS

De forma resumida, um controlador do tipo feedback é aquele que trabalha buscando corrigir a diferença entre o valor da variável de saída com o valor desejado (setpoint). O medidor do sistema lê os dados de saída e calcula o erro gerado para que, quando encaminhado ao controlador, esse atue diminuindo o offset. Se o erro for mínimo, o setpoint foi atingido, caso não, o loop será repetido até a minimização do erro (BEQUETTE, 1998).

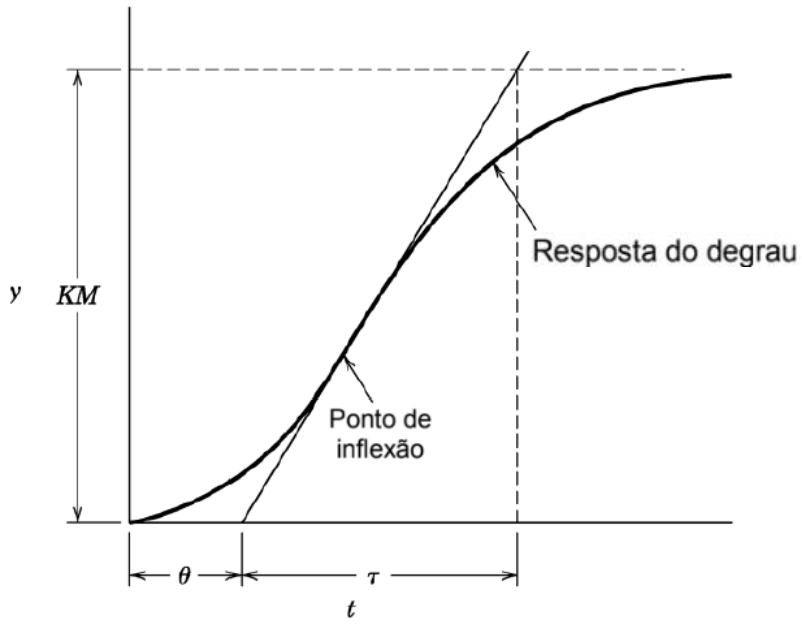
Segundo Seborg et al. (2011), a determinação da função de transferência (FT) é fundamental para o estudo da dinâmica e do controle de processo. Essa, é definida como relação entre a variável de saída e variável de entrada. A variável de saída, ou controlada, é aquela que deseja manter o valor desejado; já a variável de entrada (manipulada) é a entrada do processo que é ajustada para manter a variável controlada no valor desejado (setpoint). Com isso, tem-se a Equação 1 e a Equação 2 descrevendo como é a representação de uma FT de primeira e segunda ordem com tempo morto, respectivamente.

$$G(s) = \frac{Ke^{-\theta s}}{\tau s + 1} \tag{1}$$

$$G(s) = \frac{Ke^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \tag{2}$$

Em que *K* é o ganho do processo, θ o tempo morto e τ a constante de tempo. Uma das formas de se obter esses parâmetros é através do método gráfico ou método da curva de reação, mostrado na **Figura 1**.

Figura 1 – Gráfico representativo da curva de reação.



Fonte: adaptado de SEBORG *et al.* (2011).

O valor indicado no eixo das abcissas (*t*) representa a soma do tempo morto (θ) com a constante de tempo (τ), enquanto no eixo das ordenadas, o valor indicado por *y* é o

produto da constante do ganho do modelo (*K*) pela perturbação degrau de amplitude (*M*) na variável de entrada. Para a obtenção desses valores, traça-se uma reta tangente pelo ponto de inflexão e, em seguida, a partir da intersecção com o eixo das abcissas, tira-se o valor de θ. Além disso, tem-se que a intersecção dessa reta tangente com a linha do estado estacionário final se dá no tempo *t* ($\theta + \tau$) (ÅSTRÖM, 1995).

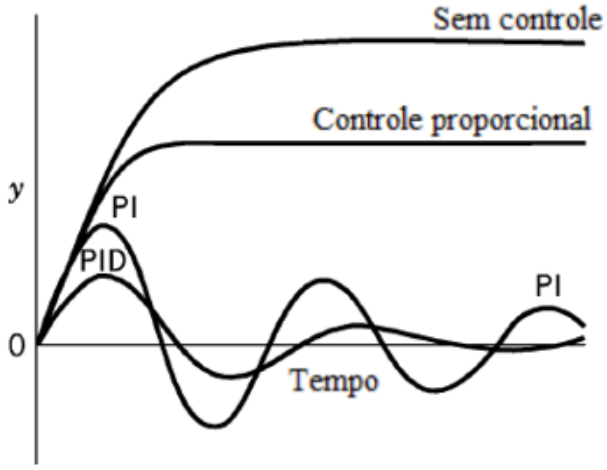
Segundo Banzella (2014), para que seja possível a obtenção da curva de reação de um sistema, deve-se realizar um controle em malha aberta, ou seja, aplica-se um sinal de entrada (degrau por exemplo) aguardando a estabilização da variável controlada num determinado valor. Porém, nesse tipo de controle, os sinais de saída não possuem influência no sinal de entrada e, portanto, não são utilizadas informações sobre a evolução do processo e nem é tomada nenhuma ação de controle efetivo na entrada, estando todo o sistema sujeito às perturbações.

Ainda de acordo com Banzella (2014), para se ter um controle em malha fechada, são utilizados sensores que medem o comportamento do processo ao longo do tempo bem como das alterações das variáveis. Esses sinais de saída são levados em consideração e direcionados à um controlador, que por sua vez vai comparar os sinais de saída do sistema com os sinais do *setpoint*. Com isso, ele tende e tomar ações a fim de diminuir o erro e controlar o mais rápido possível o sistema. Para isso, é necessário ter em mãos os parâmetros do controlador para que haja um controle efetivo. A Equação 3 demonstra os parâmetros de um controlador PID.

$$G_c(s) = K_c\left(1 + \frac{1}{\tau_I s} + \tau_D s\right) \tag{3}$$

Em que *Kc* é o parâmetro de ganho da função relacionado ao controle proporcional, τ_I e τ_D são as constantes de tempo do controle integral e derivativo, respectivamente. Esses parâmetros combinados, causam diferentes comportamentos no processo em relação ao tempo de resposta do controlador ao perceber uma mudança de *setpoint* ou uma perturbação. Tudo vai depender do ajuste fino de cada parâmetro do controlador e, também, da dinâmica do processo em que o controlador está atuando. Mas, de forma geral, as respostas típicas para sistemas de controles feedbacks é mostrado na **Figura 2**.

Figura 2 – Respostas típicas para sistemas com controle PID.



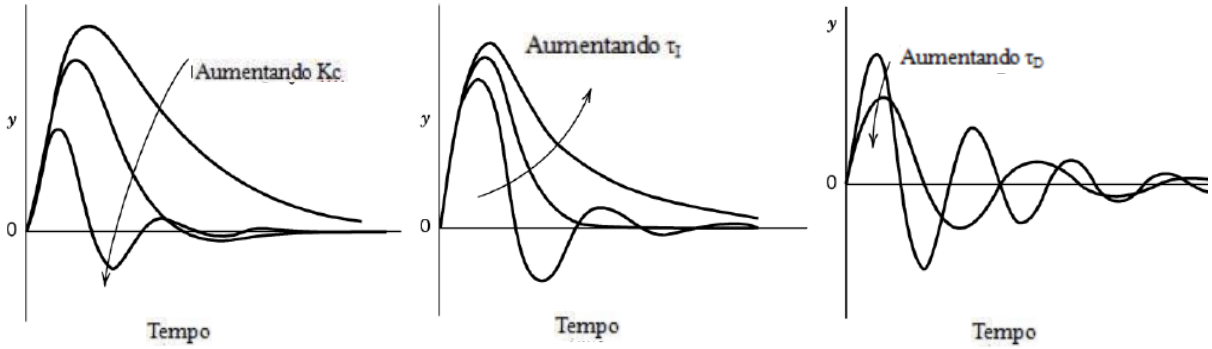
Fonte: adaptado de SEBORG *et al.* (2011).

Como já falado, um mesmo controlador por ter efeitos diferentes em um sistema simplesmente manipulando seus parâmetros proporcional, integral e derivativo. De forma resumida, o aumento do ganho do controlador (*Kc*) faz com que o sistema responda mais rapidamente, porém, se *Kc* for muito elevado, o sistema começa a gerar respostas oscilatórias e, em casos extremos, torna-se instável (SEBORG *et al.,* 2011).

Já com o aumento do tempo de integração, τ_I, os controladores assumem um comportamento mais conservador, ou seja, diminuem oscilação e tem resposta mais lenta, porém, o contrário é observado ao se aumentar o ganho do controlador. Para qualquer valor positivo adotado para τ_I, o sistema irá eliminar o *offset*, porém, a variável controlada retornará ao *setpoint* lentamente após qualquer mudança sofrida no processo (SEBORG *et al.,* 2011).

Por fim, efeito do τ_D deve ser avaliado em cada caso. Quanto se tem valores pequenos, aumentar τ_D implica uma melhora do tempo de resposta e diminui a amplitude de oscilação. Por outro lado, se τ_D for muito alto, há um efeito contrário, e o sistema se torna oscilatório. Por isso, valores intermediários de τ_D são mais indicados para controladores PID (SEBORG *et al.,* 2011). O conjunto de gráficos na **Figura 3** exemplifica o efeito de cada parâmetro no processo.

Figura 3 – Controladores P, PI e PID e o efeito de cada parâmetro no processo.



Fonte: adaptado de SEBORG *et al.* (2011).

Algoritmo de Colônia de Vagalumes

A tomada de decisões em vários campos da ciência é comumente realizada com o uso de conceitos de otimização. Essa tomada de decisão é baseada pela busca de soluções para uma determinada função, denominada Função Objetivo (FO), sendo que essa solução normalmente extremiza (valores de máximo ou mínimo) a função de referência, respeitando um determinado conjunto de restrições que variam de acordo com o problema estudado (J.D. PINTÉR, 2002).

Os algoritmos de otimização são divididos em diversos subgrupos podendo destacar os métodos determinísticos e probabilísticos. O primeiro, é caracterizado por um modelo matemático que determina os resultados, exatamente, a partir das condições iniciais, ou seja, dada uma certa entrada o mesmo apresentará uma única saída repetidamente. Já o segundo se caracteriza por permitir diferentes pontos de entrada, podendo então gerar diferentes pontos de saída, ou seja, ao contrário dos métodos determinísticos, esses não ficarão presos em uma única solução (PEREIRA, 2020).

O algoritmo utilizado para a resolução desse trabalho (Colônia de Vagalumes) se caracteriza por ser um método de otimização global com aplicações bem-sucedidas em diversos campos das ciências aplicadas. A bioluminescência tem uma influência sobre os vagalumes, já que um vagalume é atraído pelo outro devido a luminosidade que emitem. Segundo Yang (2010), existem três funções associadas a bioluminescência: ferramenta de comunicação e atração para acasalamento; isca para atração de possíveis presas; e mecanismo de alerta para potenciais predadores.

O algoritmo proposto por Yang (2010) considera as seguintes hipóteses de que os vagalumes são unissex e podem ser atraídos um pelo outro; a atratividade é diretamente proporcional a luminosidade e diminui com o aumento da distância entre eles (se não houver um vagalume mais brilhante que os demais, todos se moverão aleatoriamente); por fim, a luminosidade de um vagalume é estabelecida pela função objetivo. Se transcrito em forma de código computacional (pseudocódigo), esse seria igual ao mostrado na **Tabela 1**.

Tabela 1 – Algoritmo de Colônia de Vagalumes

Início
Definir a função objetivo *J(x)*, $x = (x_1, x_2, ..., x_D)T$
Definir os parâmetros do ACV
Para *i*=1 até o número máximo de gerações faça
Calcular a intensidade da luz *Ii* para *xi* proporcionalmente a *J(xi)*
Para *j* = 1 até o número de vagalumes
Calcular o fator de atratividade
Mover o vagalume *j* em direção aos vagalumes mais brilhantes
Verificar se o vagalume está dentro dos limites
Fim-Para
Fim-Para
Pós-processamento e visualização dos resultados
Fim

Fonte: ALMEIDA, 2021.

Em que *J(x)* corresponde a função objetivo, ($x_1, x_2, ..., x_D$) representam as variáveis de interesse, que o algoritmo fornece ao finalizar o número de iterações. Para a simulação, existe a dependência de alguns parâmetros tais como tamanho da população, número de interações, parâmetro de aleatoriedade, parâmetro de atratividade e absorção da luz.

Além disso, de acordo com Oliveira MDM et al. (2020), a variável definida como intensidade da luz é inversamente proporcional à distância entre os vagalumes. Pode-se notar pela Equação 4.

$$I = I_0 e^{-\gamma r^2} \tag{4}$$

Em que I_0 representa a intensidade inicial, *r* a distância euclidiana entre os vagalumes e γ a absorção da luz. Já a atratividade, é uma variável proporcional à intensidade da luz e representa o quanto os vagalumes estão próximos um dos outros. Essa relação está representada na Equação 5, em que β_0 representa a atratividade inicial em *r* = 0.

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{5}$$

Outra variável importante é a movimentação dos vagalumes, que representa o movimento das soluções no espaço de busca do problema. Esse movimento depende de três fatores importantes: a atratividade (β), absorção da luz pelo meio (γ) e a aleatoriedade (α). O movimento do vagalume *i* em direção ao vagalume *j* mais brilhante é definido na Equação 6.

$$x_{i+1} = x_i + \beta_0 e^{-\gamma r^2}\left(x_j - x_i\right) + \alpha\, scale(rand - 0.5) \tag{6}$$

Na equação acima temos que: o primeiro termo representa a posição atual do vagalume; o segundo termo representa a atratividade e a intensidade da luz vista pelos outros vagalumes; e, por fim, o terceiro termo indica o movimento aleatório, em que α é o fator de aleatoriedade, *rand* é o gerador de números aleatórios dentro de um intervalo [0;1] e *scale* é um vetor que garante que o vagalume esteja dentro do domínio especificado.

Arduino UNO e a Aquisição de Dados em *Pyhton*

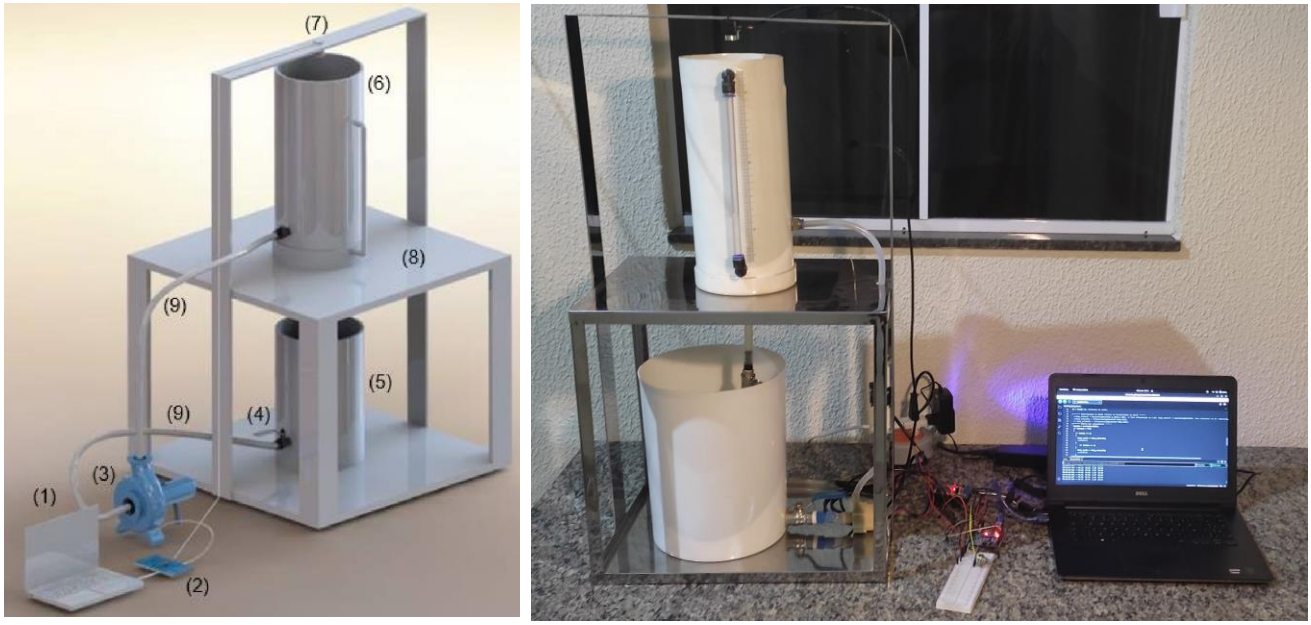
O Arduino UNO é uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, uma linguagem de programação padrão, a qual tem origem em Wiring, e é essencialmente C/C++ (WIKIPÉDIA, 2023). Seu objetivo é criar ferramentas que são acessíveis, com baixo custo, flexíveis e fáceis de se usar por principiantes e profissionais, principalmente para aqueles que não tem alcance aos controladores mais sofisticados e ferramentas mais complicadas. Pode ser usado para o desenvolvimento de objetos interativos independentes, ou ainda, para ser conectado a um computador hospedeiro.

Uma característica negativa dessa plataforma, é que os dados recebidos por ele e impressos no prompt, não ficam salvos e não há a possibilidade de salvá-los. A única forma que há de salvar os dados obtidos, seria copiando-os e salvando em outro documento. No entanto, em uma corrida com mais de dois mil pontos, essa tática acaba sendo inviável. Uma solução para esse inconveniente, foi a implementação de um código em Python que lesse os dados de saída do processo e salvasse-os em um arquivo “txt”, que pode ser usado para cálculos e iterações do algoritmo de Colônia de Vagalumes.

O *Python* é uma linguagem de programação de alto nível, ou seja, com sintaxe mais simplificada e próxima da linguagem humana, utilizada nas mais diversas aplicações, como desktop, web, servidores e ciência de dados. Foi lançado no início da década de 90 pelo programador e matemático holandês Guido Van Rossum e foi projetada para dar ênfase no trabalho do desenvolvedor, facilitando a escrita de um código limpo, simples e legível, tanto em aplicações menores quanto em programas mais complexos (MELO, 2021).

Pelo fato de a linguagem oferecer recursos como tipagem dinâmica e forte (tipo de dado do valor deve ser do mesmo tipo da variável), orientação a objetos, multiparadigmas (programação funcional e imperativa) e possibilidade de ler dados em uma porta de entrada USB, foi feito um código com a finalidade de ler os dados de saída (nível da altura da água no reservatório) e salvá-los em um arquivo “txt”. Para a realização do presente trabalho, utilizou-se na unidade experimental, representada na **Figura 4 (a) e (b)**, os seguintes itens:

Figura 4 – Unidade experimental (reservatório de líquido).



Fonte: Dos Autores, 2023.

1. Computador;
2. Microprocessador (Arduino UNO);
3. Bomba (elemento final de controle);
4. Válvula esférica manual;
5. Reservatório de líquido recirculante;
6. Reservatório para o controle de nível de líquido;
7. Medidor de nível baseado em sensor ultrassônico;
8. Suporte metálico para os reservatórios;
9. Mangueiras para condução de líquido.

Para a realização do experimento, realizou-se uma corrida na unidade experimental em malha aberta a fim de se obter a curva de reação. Para isso, foi aplicado uma perturbação do tipo pulso e, enquanto aguardava-se a estabilização do sistema, a aquisição de dados no *Python* salvou os pontos de saída em um documento *txt*. Foi usado uma perturbação do tipo pulso pelo fato de o sistema ser integrante, ou seja, uma vez alterado o *setpoint*, ele nunca mais volta para o *setpoint*. Então, foi utilizado essa perturbação para que o sistema não transbordasse conforme o nível da água fosse aumentando. Apesar da perturbação ter sido do tipo pulso, ela teve um comportamento do tipo degrau pelo fato de não ter sido considerado os dados de saída da volta do pulso.

O mesmo foi feito para o teste com os controladores, porém, ao invés de se aplicar uma perturbação no sistema, foi feito uma alteração no *setpoint*. E, de acordo com os valores das constantes proporcional, integral e derivativa colocadas no código do Arduino UNO, a aquisição de dados foi utilizada novamente para salvar os pontos de saída de cada controlador.

Para o sistema de nível, foi utilizado o Algoritmo de Colônia de Vagalumes, classificado como probabilístico e baseado no conceito de inteligência e bioluminescência dos vagalumes. Além disso, para que fosse possível executá-lo no *Scilab*, foram utilizados os seguintes parâmetros: tamanho da população como 20, número de iterações (gerações) como 250, parâmetro de aleatoriedade como 0,5 (50%) e parâmetro de atratividade como 0,2 (20%).

É importante enfatizar que esses parâmetros foram definidos por meio de testes e experimentos realizados pelos estudos de Yang (2010). Então, resolveu-se utilizar os mesmos valores utilizados pelo autor uma vez que são os mais adequados para o sistema em questão pois são testes consagrados e já testados. Além de que, alterando algum dos parâmetros, não surtiria o mesmo efeito no sistema. Por fim, nesse trabalho foi realizado a

implementação dos controladores através do computador e microcontrolador (Arduino UNO), com o objetivo de testar e validar o projeto dos controladores.

3. RESULTADOS E DISCUSSÕES

De posse dos pontos experimentais obtidos e da curva de reação, foi utilizado o algoritmo Colônia de Vagalumes no software *Scilab* para encontrar os melhores valores para K, τ e θ que melhor aproximasse o modelo teórico dos pontos experimentais. Para tal, foi usado como critério de parada o método dos mínimos quadrados, ou seja, a função objetivo é minimizar o somatório do erro ao quadrado, em que o erro é a diferença dos dados de saída com o *setpoint* do sistema.

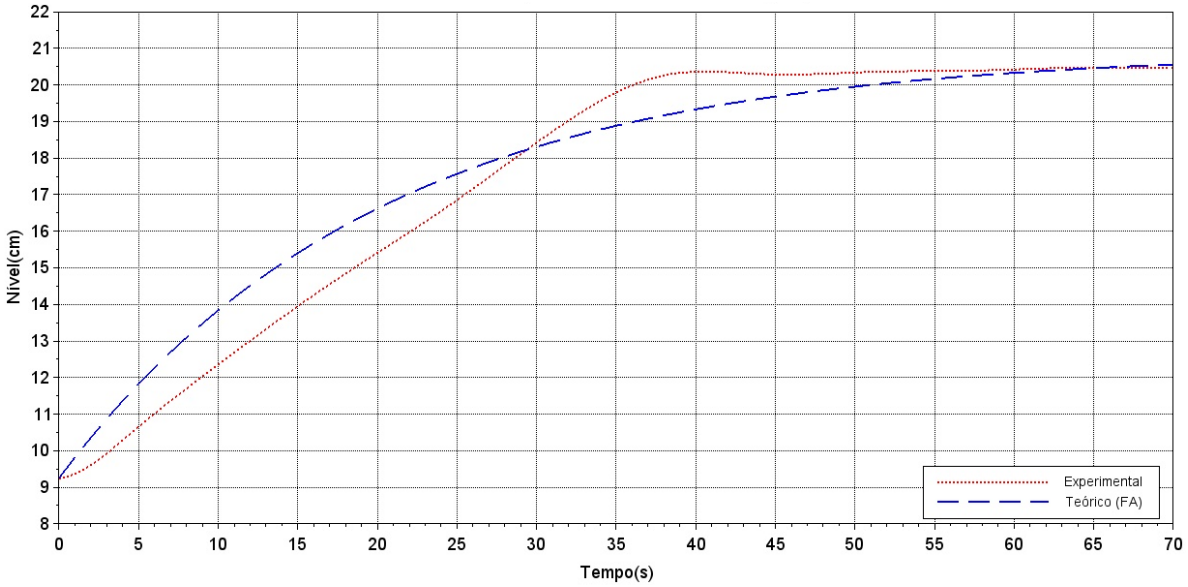
Notou-se que os cálculos realizados para o modelo de primeira ordem não foram adequados. Como observado na **Figura 5**, o erro da função objetivo foi muito elevado, conforme apresenta a **Tabela 2**, devido a não aproximação dos modelos teóricos e experimentais, evidenciando que essa ordem não é satisfatória, já que, quanto mais próximo de zero for o erro da função objetivo, melhor. Já para a função de segunda ordem, obteve-se um erro da função objetivo bem menor quando comparado com o erro do modelo de primeira ordem. Ainda, observando a **Figura 6**, pode-se notar uma melhor aproximação entre os modelos teóricos e experimentais.

Tabela 2 – Erros da função objetivo.

Modelo	Erro
Primeira ordem	1360,83
Segunda ordem	62,96

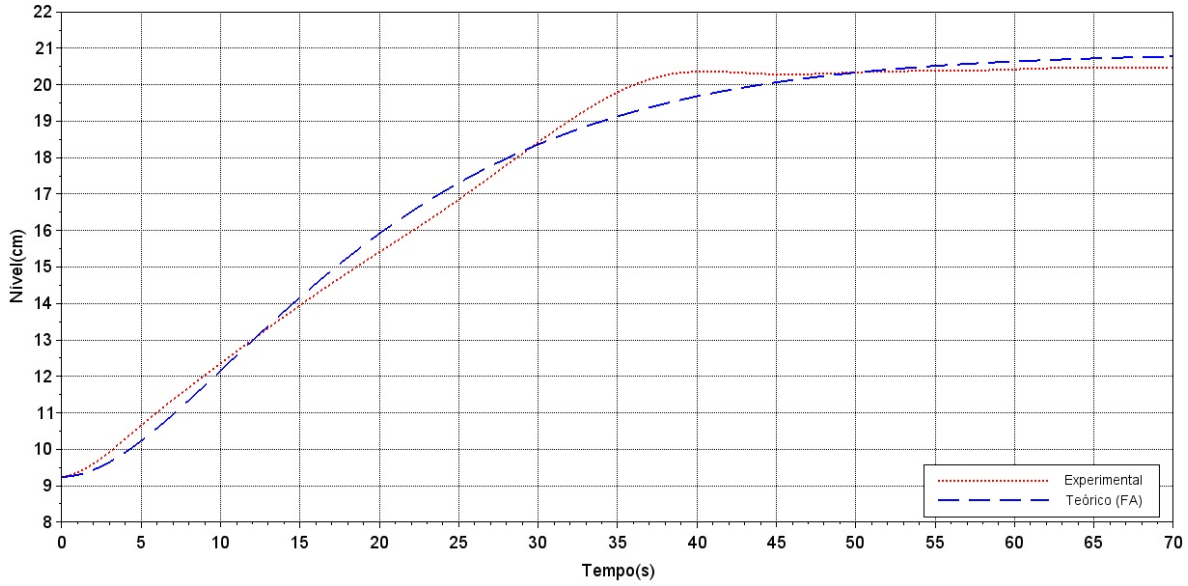
Fonte: Dos autores, 2023.

Figura 5 – Pontos experimentais vs teórico para modelo de primeira ordem.



Fonte: dos Autores, 2023.

Figura 6 – Pontos experimentais vs teórico para modelo de segunda ordem.



Fonte: Autores, 2023.

Para essa curva experimental, o algoritmo obteve os valores para os parâmetros *K*, τ_1 e τ_2 que se encontram substituídos na **Equação 2**. O valor de θ é igual a zero devido ao fato de que no sistema de nível não há tempo morto. Além disso, o formato da curva dos dados experimentais apresenta uma pequena inflexão bem como uma pequena oscilação, características essas típicas de sistemas de segunda ordem.

$$G(s) = \frac{3{,}93172}{(10{,}10130s + 1)(10{,}69270s + 1)} \quad (2)$$

Em seguida, projetou-se dois controladores usando o algoritmo Colônia de Vagalumes, um PI e outro PID. Após 250 iterações, o código gerou os seguintes valores para os parâmetros de cada controlador mostrados na **Tabela 3**.

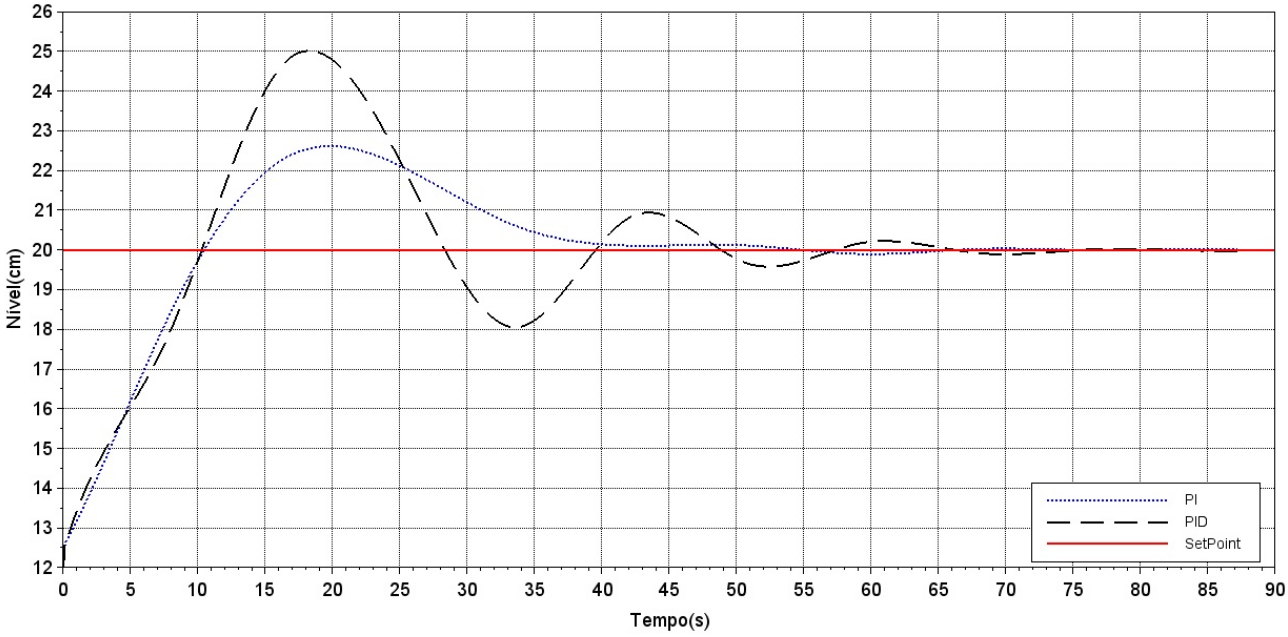
Tabela 3 – Parâmetros para os controladores PI e PID no modelo de segunda ordem.

Controlador	K_c	τ_I	τ_D
PI	10,60383	20,00000	-
PID	16,25355	5,62695	1,78710

Fonte: Dos autores, 2023.

De posse dos controladores, iniciou-se a fase de testá-los no sistema de bancada utilizado a fim de ver a atuação de cada um no processo. Para isso, foi feito uma alteração no *setpoint*: a bomba e o controlador foram ligados e foi mantido o nível de água no reservatório a uma altura de 12 centímetros; estabilizado nessa altura, mudou-se o *setpoint* para 20 centímetros e, depois, foi feita a aquisição dos dados experimentais obtidos nessa corrida. Após esse processo ser feito tanto para o PI quanto para o PID, plotou-se o gráfico de ambos os pontos experimentais respeitando a mesma quantidade de pontos para ambos os controladores. O resultado obtido está no gráfico da **Figura 7**.

Figura 7 – Pontos experimentais do comportamento dinâmico dos controladores PI e PID



Fonte: dos autores, 2023.

Como pode-se analisar, o controlador PI foi mais eficiente e mais rápido em controlar o sistema no *setpoint* em relação ao PID pelo fato de ter provocado menos oscilações. Além disso, esse fez com que o sistema atingisse o estado estacionário em aproximadamente 55 segundos, enquanto com o PID esse tempo foi de aproximadamente 75 segundos.

Para o sistema utilizando o controlador PI, foi possível observar fisicamente na planta piloto que esse controlador demandou menos do elemento final de controle (bomba), favorecendo a preservação dele. Ao utilizar o controlador PID, observou-se que para atingir o *setpoint*, esta configuração exigiu mais do elemento final de controle, provocando maior esforço no mesmo, como pode ser observado na **Figura 7**, devido às oscilações.

5. CONSIDERAÇÕES FINAIS

Portanto, no sistema utilizando o controlador PI, foi possível observar que sua atuação fez com que a estabilidade ao novo *setpoint* fosse alcançada (controlada) em aproximadamente 55 segundos e sem exigir muito do elemento final de controle, provocando menos esforço no mesmo.

Já utilizando o controlador PID para o mesmo sistema, foi possível observar que sua atuação fez com que a estabilidade ao novo *setpoint* fosse alcançada em aproximadamente 75 segundos, entretanto esse controlador exigiu muito do elemento final de controle e provocou mais esforço do mesmo. Logo, o melhor controlador para o sistema em questão foi o PI pelo fato de ter atingido o estado estacionário mais rápido e sem exigir do elemento final de controle (preservando-o) em relação ao PID.

Como sugestão para abordagens em trabalhos futuros, seria a realização de um ajuste fino nos controladores para que se chegue em uma dinâmica melhor do que a que foi obtido no presente trabalho. Além disso, a criação de uma interface amigável que funcionasse como supervisório para esse sistema em questão seria muito inovador, uma vez que faria um complemento com a aquisição de dados.

REFERÊNCIAS

ALMEIDA, Yan. **Modelagem e controle digital do nível de líquido de uma planta piloto, utilizando o algoritmo da colônia de vagalumes**. Trabalho de Iniciação Científica do Curso na Engenharia Química – UFTM, Uberaba, 2021.

Arduino. WIKIPÉDIA. Disponível em: https://pt.wikipedia.org/wiki/Arduino. Acesso em 30 de jun. 2023.

ÅSTRÖM, Karl J; HÄGGLUND, Tore. **PID Controllers: Theory, Design and Tuning**. EUA: Instrument Society of America, 1995.

BAZANELLA, Alexandre Sanfelice; JR, João Manoel Gomes da Silva. **Sistemas de Controle: princípios e métodos de projeto**. Porto Alegre: UFRGS, 2014.

BEQUETTE, B Wayne. **Process Dynamics: Modeling, Analysis and Simulation**. Primeira edição. New Jersey: Prentice Hall, 1998. p. 569.

COLOGNI, Mario. **Estudo e avaliação de metodologias de auto-sintonia de controladores PID visando uma implementação em controlador industrial**. IBICT. Florianópolis, p.1-138, 2008.

DUARTE, Grasiele Regina. **Um algoritmo inspirado em colônias de abelhas para otimização numérica com restrições**. 2015. Dissertação (Mestrado acadêmico) - Universidade Federal de Juiz de Fora, Juiz de Fora, 2015.

FACCIN, Flávio. **Abordagem Inovadora no Projeto de Controladores PID**. 2004. Dissertação de Mestrado (Pesquisa e Desenvolvimento de Processos) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004.

LOBATO, Fran Sérgio. **Otimização Multiobjetivo para o Projeto de Sistemas de Engenharia**. 2008. 402 f. Tese (Doutorado em Engenharias) - Universidade Federal de Uberlândia, Uberlândia, 2008.

LOURENÇO, João. **Sintonia de controladores PID**. Escola Superior de Tecnologia, 1997, p.12 (1-12). Disponível em: http://alvarestech.com/temp/smar/Pid.pdf. Acesso em: 25 de jun. de 2023.

MELO, Diego. Technoblog, 2021. **O que é Python? Guia para iniciantes**. Disponível em: https://tecnoblog.net/responde/o-que-e-python-guia-para-iniciantes/. Acesso em: 30 de jun. 2023.

NOGUEIRA, Marcelo. **AEC – Parte 02: Introdução aos sistemas de controle**. Disponível em: http://www.noginfo.com.br/arquivos/CC_AEC_02.pdf. Acesso em: 03 de jul. de 2023.

OLIVEIRA, Matheus. Multiobjective optimization in the level controller project in a pilot plant. **Research, Society and Development**, n. 9, p. 1-22, junho 2020.

OGATA, Katsuhiko. **Engenharia de controle moderno**, 5. ed. Pearson Prentice Hall, São Paulo, 2010.

PEREIRA, Luana *et al*. **Estudo de Sensibilidade do Algoritmo de Colônia de Vagalumes para um Problema de Engenharia Envolvendo Dimensionamento de Treliças**. Tendências em Matemática Aplicada e Computacional, 21, N. 3 (2020), 583-599. DOI: 10.5540/tema.2020.021.03.0583. Disponível em: https://www.scielo.br/j/tema/a/vDYqGbyHGN5Mb88kyhPPG8k/?lang=pt&format=pdf. Acesso em: 30 mai. 2023.

PINTER, János D. **Global Optimization: Software, Test Problems, and Applications**. In*:* Pardalos, P.M., Romeijn, H.E. (eds) Handbook of Global Optimization. Nonconvex Optimization and Its Applications, vol 62. Springer, Boston, MA. Disponível em: https://doi.org/10.1007/978-1-4757-5362-2_15. Acesso em: 01 de jul. de 2023.

SEBORG, Dale E; EDGAR, Thomas F; MELLICHAMP, Duncan A. – **Process Dynamics and Control**, 3ª edição, Wiley, 2011.

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**. nº 3, 2005, Salvador. Artigo. Salvador, 2005, pg. 1-6.

Yang, Xin-She. **Engineering Optimization: An Introduction With Metaheuristic Applications**, 1º edição. United Kingdom: John Willey and Sons, 2010. p. 347.