

Algoritmo Genético para Otimização de Controlador PID

André Thiessen, Leonardo Peres, José Ronaldo

Divisão de Engenharia de Computação, Instituto Tecnológico de Aeronáutica

Abstract—Este projeto tem como objetivo comparar o desempenho de dois métodos distintos para ajuste de controladores PID: o Algoritmo Genético (AG) e o método de Ziegler-Nichols. Métodos tradicionais como o Ziegler-Nichols fornecem um ajuste inicial rápido baseado em regras empíricas, mas frequentemente necessitam de ajustes adicionais. Por outro lado, técnicas de otimização computacional como AG exploram o espaço de busca de forma mais abrangente, potencialmente oferecendo um desempenho melhor em sistemas complexos e não lineares. Este estudo envolve a implementação de cada método de ajuste, a realização de simulações e a avaliação dos resultados com base em gráficos de resposta pelo tempo. O objetivo é fornecer uma comparação detalhada que destaque as forças e fraquezas de cada abordagem, contribuindo para uma tomada de decisão mais informada em aplicações de controle industrial.

I. INTRODUÇÃO

OS controladores Proporcionais, Integrais e Derivativos (PID) têm sido amplamente utilizados em aplicações de controle industrial devido à sua simplicidade, robustez e eficácia no controle de sistemas dinâmicos. Desde sua introdução no final do século XIX, os controladores PID se tornaram a escolha predominante em cerca de 90% das aplicações de controle de processos industriais, conforme uma pesquisa realizada em 1989. Esta popularidade é atribuída à sua facilidade de implementação e capacidade de ajuste em linha, permitindo que os engenheiros de controle otimizem o desempenho do sistema em tempo real.

A sintonização (tuning) dos parâmetros do controlador PID (ganhos proporcional, integral e derivativo) é uma tarefa crucial para garantir a performance desejada do sistema de controle. Tradicionalmente, diversas técnicas clássicas de sintonização têm sido empregadas, como o Método de Ziegler-Nichols, que data de 1942. Estas técnicas geralmente assumem um modelo simplificado da planta e derivam as configurações do controlador baseadas em características específicas da resposta do sistema, como a resposta a entrada degrau. Embora sejam rápidas e fáceis de implementar, as técnicas clássicas frequentemente exigem ajustes posteriores, pois podem resultar em respostas agressivas e oscilações excessivas.

Em contraste, as técnicas computacionais ou de otimização, como os Algoritmos Genéticos, utilizam modelos matemáticos e métodos de otimização para ajustar os parâmetros do controlador PID. Estas técnicas não fazem suposições simplificadas sobre a planta, mas em vez disso, exploram um espaço de busca mais amplo para encontrar configurações ótimas dos parâmetros. Algoritmos como os Genéticos simulam processos evolutivos naturais para encontrar soluções de alta qualidade,

oferecendo potencial para melhor performance em sistemas complexos e não lineares.

Este projeto tem como objetivo principal comparar a performance do Algoritmo Genético e do Método de Ziegler-Nichols na sintonização de um controlador PID, descobrindo as vantagens e desvantagens de cada abordagem.

II. FUNDAMENTAÇÃO TEÓRICA

A. Controladores PID

Os controladores Proporcionais, Integrais e Derivativos (PID) são amplamente utilizados em sistemas de controle devido à sua simplicidade e eficácia. O controlador PID é nomeado assim porque sua saída é a soma de três termos: proporcional (P), integral (I) e derivativo (D). Cada um desses termos é baseado no erro $e(t)$, que é a diferença entre a entrada desejada e a saída real do sistema.

A equação de um controlador PID é dada por:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

onde:

- K_p é o ganho proporcional,
- K_i é o ganho integral,
- K_d é o ganho derivativo.

Cada um desses termos influencia a resposta do sistema de maneira diferente:

- Termo Proporcional (P): Acelera a resposta do sistema, reduzindo o tempo de subida, mas não elimina o erro em regime permanente.
- Termo Integral (I): Elimina o erro em regime permanente ao aumentar a ordem do sistema, mas pode introduzir oscilações.
- Termo Derivativo (D): Reduz a magnitude das oscilações, melhorando a estabilidade do sistema, mas não afeta o erro em regime permanente.

Dessa forma, com um controlador PID, conseguimos manter uma condição ($u(t)$) desejada, corrigindo erros entre a condição real e a desejada.

A sintonia (tuning) PID é o processo de ajuste das constantes de forma a otimizar o desempenho do sistema. Este processo é crucial para garantir que o sistema responda de maneira desejada às alterações e erros.

B. Método de Ziegler-Nichols

O Método de Ziegler-Nichols é uma abordagem empírica popular para a sintonização de controladores PID, amplamente utilizado por sua simplicidade e eficácia em fornecer uma resposta inicial rápida, mesmo para aqueles sem formação teórica em controle. Este método é baseado na experimentação prática e em heurísticas para ajustar os ganhos do controlador PID de forma iterativa.

O processo de sintonização pelo Método de Ziegler-Nichols pode ser descrito pelos seguintes passos:

- 1) Configuração Inicial: Iniciar com $K_p = K_i = K_d = 0$.
- 2) Ajuste do Ganho Proporcional: Aumentar gradualmente o ganho proporcional K_p até que o sistema alcance oscilações sustentadas, ou seja, atinja o limiar de estabilidade.
- 3) Determinação dos Parâmetros Críticos:
 - Registrar o valor de K_p que causou oscilações sustentadas como K_u (ganho final ou ultimate gain).
 - Medir o período das oscilações T_u .

Com os valores de K_u e T_u determinados, os parâmetros do controlador PID são ajustados conforme as regras apresentadas na Tabela 1:

Tipo do Controlador	K_p	T_i	T_d
P	$0,5K_u$	-	-
PI	$0,45K_u$	$T_u/1,2$	-
PID	$0,6K_u$	$T_u/2$	$T_u/8$

TABLE I

REGRAS PARA AJUSTE DE PARÂMETROS DE ACORDO COM O MÉTODO DE ZIEGLER-NICHOLS

A expressão do controlador PID, conforme assumida por Ziegler-Nichols, é dada por:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right]$$

onde K_i e K_d são relacionados a K_p , T_i e T_d pelas seguintes relações:

$$K_i = \frac{K_p}{T_i}, \quad K_d = K_p T_d$$

C. Algoritmos Genéticos

Os Algoritmos Genéticos (GA) são métodos de otimização baseados nos princípios da evolução natural, como seleção, cruzamento e mutação. Eles são utilizados para resolver problemas de otimização, incluindo a sintonização de controladores PID.

O processo de um Algoritmo Genético envolve os seguintes passos:

1. Inicialização: Uma população inicial de soluções é gerada aleatoriamente.
2. Seleção: Durante cada iteração, uma proporção das soluções é selecionada com base na função de aptidão, favorecendo soluções mais “aptas”.

3. Reprodução: As soluções selecionadas são combinadas usando operações de cruzamento e mutação para criar a próxima geração de soluções.

4. Término: O algoritmo termina quando a condição de parada é alcançada, seja um número de iterações ou uma solução suficientemente boa.

O GA busca minimizar uma função de custo que pode ser baseada em critérios como Erro Absoluto Integral (IAE), Erro Quadrático Integral (ISE), Erro Absoluto Integral do Tempo (ITAE) e Erro Quadrático Integral do Tempo (ITSE).

$$IAE = \int_0^T |e(t)| dt$$

$$ISE = \int_0^T e(t)^2 dt$$

$$ITAE = \int_0^T t|e(t)| dt$$

$$ITSE = \int_0^T te(t)^2 dt$$

III. METODOLOGIA

1) *Descrição da Implementação*: O projeto implementa algoritmos de controle PID (Proporcional-Integral-Derivativo) utilizando duas abordagens distintas: Algoritmo Genético e Método de Ziegler-Nichols. O objetivo principal é determinar os melhores parâmetros PID para o controle de um sistema dinâmico, otimizando sua resposta ao degrau. A implementação foi desenvolvida em Python, uma linguagem de programação amplamente utilizada para aplicações científicas e de engenharia. As principais bibliotecas utilizadas incluem:

- **numpy**: Para operações numéricas eficientes e manipulação de arrays.
- **scipy**: Para funções científicas e técnicas, incluindo a simulação de sistemas dinâmicos.
- **matplotlib**: Para a visualização dos resultados através de gráficos.
- **control**: Para modelagem e análise de sistemas de controle.
- **tqdm**: Para criação de barras de progresso.

2) *Manual do Usuário*: Para executar o código, siga estas etapas:

• Obtenha o repositório:

- 1) Acesse o repositório no GitHub: <https://github.com/leopers/genetic-PID-tunning>.
- 2) Faça um clone do repositório para sua máquina local utilizando o comando:

```
git clone https://github.com/leopers/genetic-PID-tunning
```

• Instale as dependências:

- 1) Navegue até o diretório do projeto clonado:


```
cd genetic-PID-tunning
```

- 2) Utilize o arquivo `requirements.txt` para instalar todas as dependências necessárias:

```
pip install -r requirements.txt
```

• **Personalize os parâmetros do experimento:**

- No arquivo `main.py`, você pode ajustar os seguintes parâmetros:
 - 1) **Função de transferência do sistema:** Defina os coeficientes do numerador (`num`) e denominador (`den`) para modelar o sistema desejado.
 - 2) **Hiperparâmetros:** Ajuste os hiperparâmetros do algoritmo genético: número de bits por variável, limites para os ganhos, tamanho da população e *threshold* de *fitness*.
 - 3) K_{pu} : Ganho proporcional crítico para o método de Ziegler-Nichols. (Esse parâmetro depende da planta utilizada).

• **Execute o código:**

- 1) Navegue até a pasta `src` dentro do diretório do projeto:

```
cd src
```

- 2) Execute o arquivo `main.py` para iniciar a simulação:

```
python main.py
```

IV. RESULTADOS E DISCUSSÕES

Para análise da implementação realizada, foi escolhida a função de transferência de malha aberta:

$$G(s) = \frac{20}{s^3 + 32s^2 + 140s}$$

A partir, daí, os hiperparâmetros escolhidos para o algoritmo genético de otimização estão apresentados na Tabela II.

TABLE II
PARÂMETROS DO ALGORITMO GENÉTICO PARA TUNING DO
CONTROLADOR PID

Parâmetro	Valor
Número de Variáveis (<code>n_var</code>)	3 (K_p , K_i , K_d)
Número de Bits por Variável (<code>n_bit</code>)	5
Limite Superior (<code>ra</code>)	100
Limite Inferior (<code>rb</code>)	0
Tamanho da População (<code>population</code>)	100
Alvo Mínimo (<code>minimum_target</code>)	83

Além disso, a função de fitness é projetada para avaliar o desempenho de um controlador PID com base em várias métricas de resposta ao degrau. As variáveis de interesse são:

- *Rise Time* (t_r): O tempo necessário para a resposta alcançar pela primeira vez um valor específico.
- *Steady-State Error* (ess): O erro de regime permanente.
- *Overshoot* (M_p): A quantidade que o sinal de saída excede seu valor de regime permanente.
- *Settling Time* (t_s): O tempo necessário para a resposta permanecer dentro de uma banda específica ao redor do valor final.

A função de fitness considera as seguintes métricas:

$$Fitness_1 = \frac{100}{t_r + 1} \quad (1)$$

$$Fitness_2 = \frac{100}{ess + 0.1} \quad (2)$$

$$Fitness_3 = \begin{cases} \frac{100}{M_p + 1} & \text{se } ess \neq 0 \\ 100 & \text{se } ess = 0 \end{cases} \quad (3)$$

$$Fitness_4 = \begin{cases} \frac{100}{t_s + 0.01} & \text{se } t_s > 10 \\ 100 & \text{se } t_s \leq 10 \end{cases} \quad (4)$$

A função de fitness final é calculada como a média aritmética das métricas individuais:

$$Fitness = \frac{Fitness_1 + Fitness_2 + Fitness_3 + Fitness_4}{4} \quad (5)$$

Assim, pode-se observar as evoluções de cada um dos ganhos do controlador PID ao longo das gerações do algoritmo:

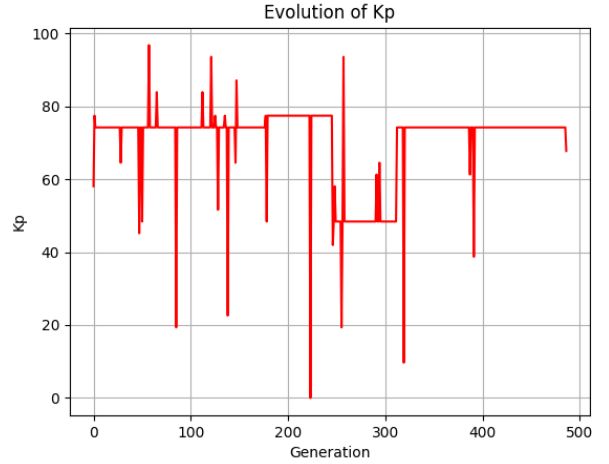


Fig. 1. Evolução de K_p

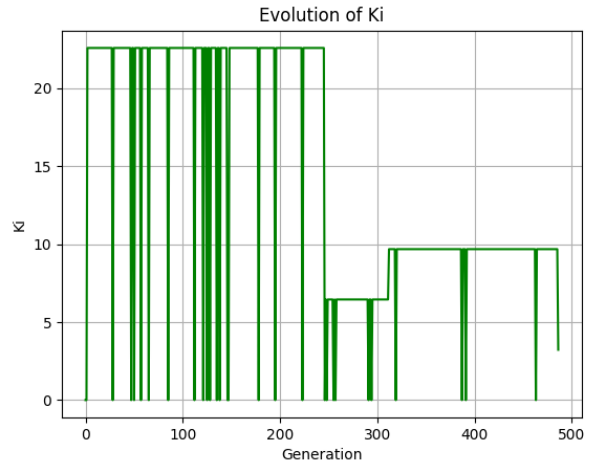


Fig. 2. Evolução de K_i

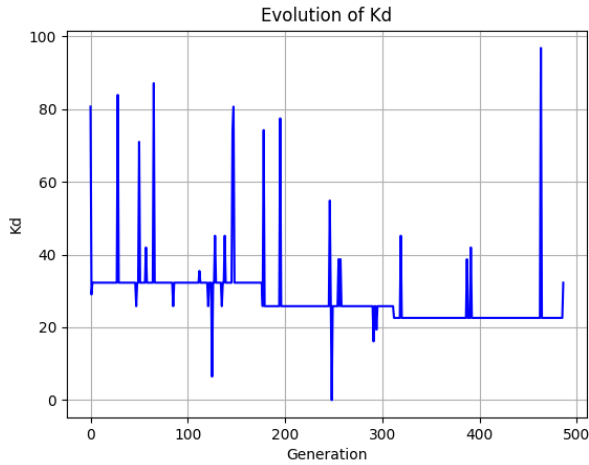
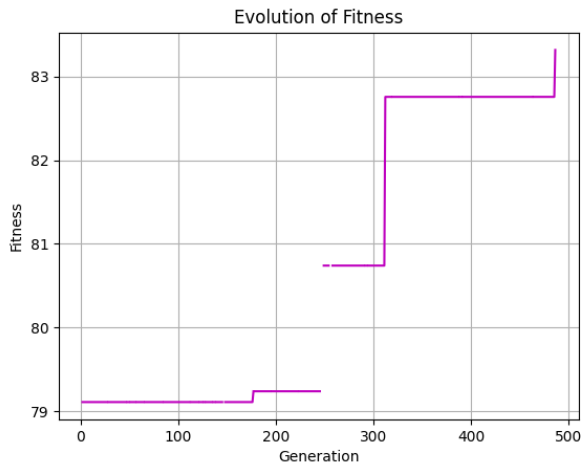


Fig. 3. Evolução de Kd

Além disso, podemos observar como a função de *fitness* cresce até que o *threshold* seja atingido:

Fig. 4. Evolução do *fitness*

Em contrapartida, para a sintonização do controlador pelo método de Ziegler-Nichols, foi necessário se calcular analiticamente o ganho proporcional que deixa o sistema de malha fechada no limiar de estabilidade (K_u). Para isso, utilizou-se a resposta em frequência do sistema de malha fechada para se calcular o ganho para o qual se obtém o limiar de estabilidade.

$$G_f(s) = \frac{KG(s)}{1 + KG(s)} = \frac{20K}{s^3 + 32s^2 + 140s + 20K} = \frac{N_f(s)}{D_f(s)}.$$

$$\begin{aligned} D_f(j\omega) &= (j\omega)^3 + 32(j\omega)^2 + 140j\omega + 20K = 0 \\ \Rightarrow \begin{cases} -\omega^3 + 140\omega = 0 \\ -32\omega^2 + 20K = 0 \end{cases} &\Rightarrow \begin{cases} \omega = 11.83 \text{ rad/s} \\ K_u = 224 \end{cases} \end{aligned}$$

Assim, utilizando a regra de Ziegler-Nichols pôde-se sintonizar os ganhos do controlador PID e comparar o de-

sempenho dessa metodologia com a otimização pelo uso de Algoritmo Genético.

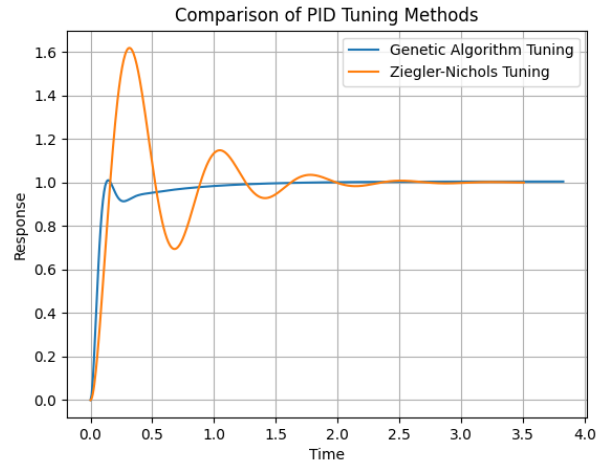
Fig. 5. Comparação da resposta ao degrau para ambos os *tunings*

TABLE III
COMPARAÇÃO DOS PARÂMETROS PID

Parâmetro	GA	ZN
Kp	67.742	134.400
Ki	3.226	505.733
Kd	32.258	8.929
Tempo de Subida (s)	0.073	0.112
Overshoot (%)	1.033	61.805
Tempo de Acomodação (s)	0.915	1.900
Erro em Regime Permanente	0.000	0.000

A partir dos resultados apresentados na Tabela III, pode-se perceber que a estratégia numérica, representada pelo Algoritmo Genético (GA), consegue desempenhar consideravelmente melhor do que o método de Ziegler-Nichols (ZN). O controlador sintonizado pelo GA apresenta uma resposta mais rápida, com um tempo de subida significativamente menor, uma resposta mais amortecida, com overshoot quase inexistente, e menores valores de ganhos proporcional (Kp), integral (Ki) e derivativo (Kd).

Esses resultados indicam que a estratégia numérica é mais eficaz em termos de desempenho dinâmico do sistema, proporcionando uma resposta mais estável e rápida. No entanto, é importante considerar algumas desvantagens dessa estratégia de otimização. Primeiramente, o GA pode ser susceptível a convergir para mínimos locais, o que significa que ele pode não encontrar a solução globalmente ótima em alguns casos. Isso pode levar a um desempenho subótimo do controlador em situações onde a superfície de busca da função de *fitness* é altamente não linear e multimodal.

Além disso, o custo computacional associado ao GA é significativamente alto. A execução do algoritmo pode exigir um grande número de avaliações da função de *fitness*, especialmente em sistemas complexos com muitas variáveis de controle. Isso pode resultar em tempos de execução prolongados e alto consumo de recursos computacionais, o que pode

ser uma limitação em aplicações que requerem respostas em tempo real ou em sistemas com restrições de processamento.

Apesar dessas desvantagens, a superioridade do desempenho dinâmico obtido pelo GA pode justificar seu uso em aplicações onde a precisão e a qualidade da resposta do controlador são cruciais. Em resumo, a escolha entre o método numérico e o método tradicional de Ziegler-Nichols deve considerar o equilíbrio entre a necessidade de desempenho superior e os recursos computacionais disponíveis.

V. CONCLUSÃO

Através da comparação detalhada entre os métodos de sintonização de controladores PID utilizando Algoritmos Genéticos (GA) e o método tradicional de Ziegler-Nichols (ZN), foi possível observar diferenças significativas em termos de desempenho do sistema.

Os resultados mostraram que a estratégia numérica baseada em Algoritmos Genéticos apresenta uma resposta ao degrau mais rápida e amortecida, com menores valores de overshoot e tempo de acomodação. Esses benefícios destacam a eficácia do GA em otimizar o desempenho dinâmico do controlador PID, especialmente em sistemas complexos e não lineares.

Por outro lado, as desvantagens da utilização de GA incluem a possibilidade de convergência para mínimos locais e o alto custo computacional, devido ao grande número de avaliações da função de fitness necessárias para alcançar uma solução otimizada. Esses fatores podem limitar a aplicação do GA em sistemas que exigem respostas em tempo real ou possuem restrições de recursos computacionais.

A escolha entre o método de Ziegler-Nichols e os Algoritmos Genéticos deve, portanto, considerar um equilíbrio entre a necessidade de um desempenho superior e a disponibilidade de recursos computacionais. Enquanto o método de Ziegler-Nichols oferece uma solução rápida e empírica, o GA proporciona uma otimização mais robusta e detalhada, apesar de seu maior custo computacional.

Em suma, a estratégia numérica baseada em Algoritmos Genéticos demonstrou um desempenho consideravelmente melhor em termos de tempo de resposta e estabilidade do sistema, justificando seu uso em aplicações onde a precisão e a qualidade da resposta do controlador são essenciais. No entanto, a escolha do método de sintonização deve ser feita com base nas necessidades específicas da aplicação e nos recursos disponíveis.

AGRADECIMENTO

Gostaríamos de expressar nossa profunda gratidão ao professor Marcos Máximo pelas aulas de CMC-12 ministradas ao longo do semestre e pelo apoio contínuo. Sua dedicação em proporcionar o melhor ensino da disciplina foi essencial para nosso aprendizado e, conseqüentemente, para o desenvolvimento deste projeto de exame.