ALGORITMO DETERMINÍSTICO PARA LIMITAÇÃO DO MÁXIMO SOBRESSINAL EM SISTEMAS DE CONTROLE PID QUANDO SINTONIZADOS PELO MÉTODO DA SÍNTESE DIRETA

ANDRADE FILHO, A. P[1,2], TEIXEIRA, E. P[2], SILVA, A. M. B[2]

[1] Universidade Federal de Uberlândia – FEELT Faculdade de Engenharia Elétrica
[2] Universidade de Uberaba – Mestrado Profissional em Engenharia Química

RESUMO – *Nas plantas de controle industriais, a maioria dos controladores PID são ajustados em campo, apesar de existirem diversos métodos de sintonia destes controladores. Sem dúvida, o método da síntese direta é um dos mais eficazes e simples de se implementar, ainda mais quando não se conhece a função de transferência característica do processo. Entretanto, para se implementar este método, é necessário arbitrar um valor para a constante de tempo desejada do sistema. O problema é que, quanto menor for esta constante de tempo, o sistema tenderá a apresentar um sobressinal mais acentuado, o que não é aceitável em muitos sistemas de controle. O presente artigo apresenta um algoritmo implementado na linguagem VBA para calcular e determinar os parâmetros do PID (Proporcional Integral Derivativo) de forma automática, baseando-se em uma constante de tempo ideal, que após determinada pelo algoritmo gerará a melhor resposta abaixo do sobressinal desejado e informado pelo usuário.*

1. INTRODUÇÃO

Em muitas aplicações, os controladores PID são utilizados de forma ineficiente pois são mal sintonizados. No contexto da indústria química, o desafio é ainda maior pois requer malhas PID bem sintonizadas, além disso, os processos químicos são naturalmente multivariáveis e com interações significantes entre as suas variáveis de entrada e de saída, levando o engenheiro de processos, responsável por centenas de malhas, a necessitar de uma estratégia de sintonia de fácil implantação, mas necessariamente robusta. O presente artigo propõe uma abordagem mais assertiva da sintonia dos controladores PID quando sintonizados pelo método da síntese direta, uma vez que de forma convencional, deve-se arbitrar uma constante de tempo desejada, para se obterem os parâmetros proporcional, integral e derivativo ideais. Isto deixa o método pouco eficiente no que se refere ao controle dos demais parâmetros de projeto, pois quando queremos que um sistema responda de forma mais rápida, a constante de tempo arbitrada deve ser pequena, o problema é que com uma constante de tempo muito diminuta ocorre um aumento no sobressinal, o que é de fato um problema inaceitável para diversas plantas de controle. Sendo assim, o presente artigo propõe um algoritmo e o implementa na linguagem VBA para calcular e determinar os parâmetros do PID de forma automática, baseando-se em uma constante de tempo ideal, que após determinada pelo algoritmo gerará a melhor resposta abaixo do sobressinal desejado e informado pelo operador do processo.

2. REVISÃO DA LITERATURA

Controladores PID são especialmente úteis devido a ação conjunta das ações proporcional, integral e derivativa (Nise,2002), sendo que em relação a ação proporcional, atua de forma direta e imediata no erro de controle. Além do mais, a ação proporcional não leva em consideração o desempenho passado do sistema, não se baseia na tendência de evolução do processo e apresenta erro de regime permanente na maioria dos casos. A sua principal vantagem é a simplicidade de implementação. A ação Integral, por sua vez, atua de forma cumulativa em cada instante de amostragem considerando o incremento da área entre o erro de controle e o eixo horizontal. Ele é acumulado em uma posição de memória. Desta forma, a integral é calculada por meio do cálculo cumulativo da área. Passado o efeito da perturbação o saldo da integral contribui na determinação do valor final da variável de controle (Ogata,2004). A parcela integral tem como principal finalidade a eliminação do erro de regime permanente. A ação derivativa deve-se ao fato de a derivada indicar a tendência de variação de uma função que, neste caso, é o sinal de entrada.

Considere a equação do controlador PID independente:

$$m(t) = Pe(t) + I\int_0^t e(t)dt + D\frac{de(t)}{dt} \qquad [1]$$

Aplicando-se Laplace, considerando-se o sistema em repouso, ou seja, com as condições iniciais nulas, obtém-se:

$$M(s) = P\,E(s) + \frac{I}{s}\,E(s) + D\,s\,E(s) \qquad [2]$$

Tomando-se a relação entre a entrada e a saída, obtém-se a função de transferência do controlador:

$$G_c(s) = \frac{M(s)}{E(s)} = P + \frac{I}{s} + Ds = \frac{Ps + I + Ds^2}{s} \qquad [3]$$

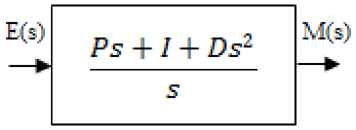

Figura 1 – Função de transferência do controlador PID

3. MATERIAIS E MÉTODOS

3.1 Dedução da equação de sintonia do controlador PID pela síntese direta

Considerando o sistema representado abaixo:

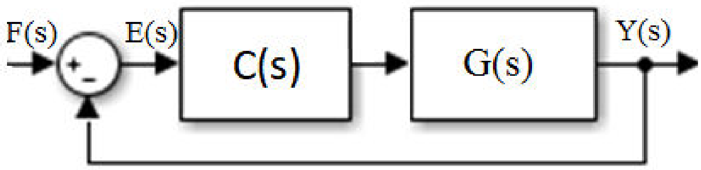

Figura 2 – Diagrama de processo genérico.

Considere $G_d(s)$ como a função de transferência de malha fechada (Teixeira,2020), após a aplicação do controlador C(s). Desta forma, deseja-se que, em malha fechada, tenha-se:

$$G_{eq}(s) = G_d(s) \qquad [4]$$

$G_{eq}(s)$: Função de transferência equivalente;

$$G_{eq}(s) = \frac{C(s)G(s)}{1 + C(s)\,G(s)} = G_d(s) \qquad [5]$$

Considere também que o processo possa ser representado por uma função de transferência de primeira ordem com tempo.

$$Considere: G(s) = \frac{k\,e^{-\theta s}}{\tau s + 1} \qquad [6]$$

Onde k é o ganho do processo, θ é tempo morto e τ é a constante de tempo do processo.

$$Deseja - se: G_d(s) = \frac{1\,e^{-\theta s}}{\tau_d s + 1} \qquad [7]$$

Pois, com ganho igual a 1, não haverá erro de regime permanente.

A incógnita em (5) é C(s). Isto é, deseja-se ajustar o controlador para que: $G_{eq}(s) = G_d(s)$

Em (5):

$$C(s)G(s) = G_d(s)[1 + C(s)G(s)] \quad ou \ \rightarrow C(s)G(s)[1 - G_d(s)] = G_d(s)$$

$$C(s) = \frac{G_d(s)}{G(s)[1 - G_d(s)]} \qquad [8]$$

Substituindo (6) e (7) em (8):

$$C(s) = \frac{\frac{1e^{-\theta s}}{\tau_{d(s)} + 1}}{\frac{k\ e^{-\theta s}}{\tau s + 1}\left[1 - \frac{e^{-\theta s}}{\tau_d s + 1}\right]} \rightarrow C(s) = \frac{\tau s + 1}{k[\tau_d s + 1 - e^{-\theta s}]} \qquad [9]$$

Pela série de Maclaurin $e^{-\theta s} \approx 1 - \theta s$, usando -se somente o primeiro termo da série tem-se:

$$C(s) = \frac{\tau s + 1}{k[\tau_d s + 1 - 1 + \theta s]}$$

$$C(s) = \frac{\tau s + 1}{k(\tau_d + \theta)s} \rightarrow C(s) = \frac{\tau s}{k(\tau_d + \theta)s} + \frac{1}{k(\tau_d + \theta)s}$$

$$Considere\text{: } P = \frac{\tau}{k(\tau_d + \theta)} \qquad [10] \qquad e \qquad I = \frac{1}{k(\tau_d + \theta)} \qquad [11]$$

3.2 Método da Síntese Direta Clássico

O método de sintonia por síntese direta clássico consiste em se arbitrar uma *constante de tempo desejada* (τ_d), que aliada aos parâmetros da função de transferência *tempo morto ou atraso (Θ), ganho (k)* e *constante de tempo da função de transferência* (τ), seja possível através das equações demonstradas em 3.1 obter rapidamente os valores P e I do controlador PI. Se considerarmos os parâmetros da função de transferência sendo: $\Theta = 0{,}5$; $K = 0{,}8$; $\tau = 2$; e arbitrando $\tau_d = 0{,}1$, tem-se, por meio das equações (10) e (11), os seguintes resultados: P = 4,16 e I = 2,08. Simulando o sistema no Scilab com os valores P e I calculados, tem-se a resposta apresentada na figura 3.1. Entretanto, como se pode notar, o sobressinal foi demasiadamente alto, chegando a ultrapassar 33,44% do setpoint, o que em muitos sistemas não é aceitável.

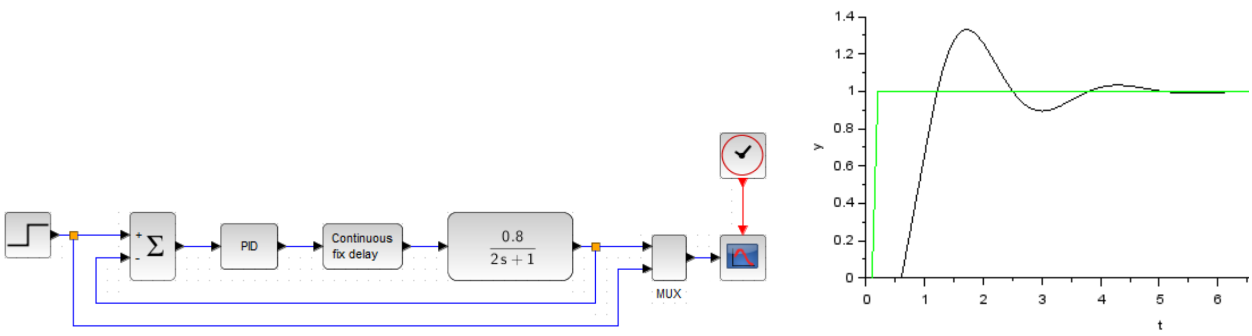

Figura 3 – Resposta do sistema com τ_d arbitrado em 0,1.

Para que se possa obter um resultado otimizado, apresenta-se o método automático de sintonia proposto neste trabalho.

4. RESULTADOS E DISCUSSÃO

4.1 Aplicação da Síntese Direta Automatizada

O método da *Síntese Direta Otimizado* proposto neste artigo, consiste em um algoritmo que recebe do operador do sistema o valor máximo desejável do sobressinal e calcula a resposta, variando-se τ_d entre o intervalo de 0.01 até a constante de tempo da função de transferência do sistema (τ). Para cada valor de τ_d a ser considerado na análise de otimização, os valores discretizados da saída (PV) e o valor de cada τ_d correspondente ao intervalo amostrado, sendo T = 0,1, são inseridos automaticamente em uma matriz de dados Double 2x300 denominada *pv_array( ). O*u seja, em 30 segundos de amostragem são armazenados 300 pontos discretizados do sinal analisado (Dorf,2001). Sendo assim, após o processo de simulação serão geradas N matrizes, que para o escopo desta simulação estão limitadas em 200, pois por simplicidade, optou-se por limitar a constante de tempo do sistema (τ) para simulação igual ao valor do τ da função de transferência utilizada no exemplo, 2. Entretanto esse parâmetro pode facilmente ser alterado no algoritmo utilizando um vetor dinâmico, fazendo assim com que a simulação seja possível para respostas além da constante de tempo do sistema.

Na etapa seguinte, é realizado um processo de leitura da matriz pv_array() onde se buscam todos os valores menores ou iguais aos determinados pelo usuário do sistema. Cada par (τ_d, PV) é então armazenado em uma outra matriz denominada pv_array_aux(), onde posteriormente também será analisada afim de procurar o PV e consequentemente τ_d mais próximo do estipulado pelo usuário. No passo seguinte, os valores são inseridos em uma planilha do Excel onde está implementado o equacionamento através da e*quação de diferença* da função de transferência do sistema que é então plotada, a fim de que seja possível avaliar graficamente a resposta de saída gerada pelo sistema. Nesta mesma planilha, o algoritmo retorna o τ_d ideal que por sua vez gera os parâmetros P e I para que se obtenha o valor de sobressinal estipulado pelo usuário do sistema. A figura 4 apresenta o resultado da aplicação do algoritmo para um máximo sobressinal exigido pelo usuário de 5%:

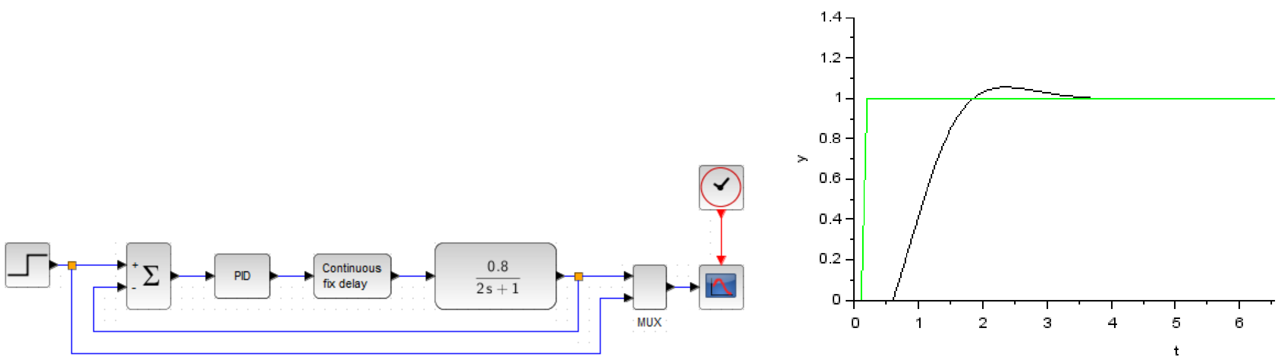

Figura 4 – Resposta do sistema → τ_d=0,4

Resultado calculado pelo algoritmo: τ_d = 0,4 → P = 2,77, I = 1,38, sobressinal = 4,69%.

A figura 5 apresenta o trecho de código que está dentro do laço que calcula o resultado do PID simulado para cada uma das 200 amostras discretizadas:

```
For j = 1 To 200

    'Calcula P-I pelo método da Síntese direta
    p = (tau / (k * (taud + teta)))
    i = (1 / (k * (taud + teta)))
    Range("J2").Value = p
    Range("K2").Value = i

    'Escreve os valores da coluna PV no array "pv_array" e coloca o maior valor dentro da variável max_po
    For n = 1 To 300
        pv_array(n) = Range("D" & (n + 1)).Value
        If n = 1 Then
            max_po = pv_array(1)
        ElseIf pv_array(n) > max_po Then
            max_po = pv_array(n)
        End If
        max_taud = taud
    Next n
    'Colocar os valores máximos na matriz de comparação
    pv_matriz(1, j) = max_po
    pv_matriz(2, j) = max_taud

    'Zera as variáveis para análise matriz seguinte
    max_po = 0
    max_taud = 0

    'salvar na matriz o taud correspondente ao valor máximo
    taud = taud + 0.01

Next j
```

Figura 5 – Trecho do algoritmo para o cálculo do τ_d ideal

5. CONCLUSÃO

Como resultado do processo de desenvolvimento deste artigo, ficou demonstrado que o *algoritmo* proposto é capaz de limitar o sobressinal determinando, considerando-se um tempo de acomodação máximo estipulado pelo usuário.

Além do êxito em se atingir a proposta do trabalho, obteve-se paralelamente em decorrência do processo de pesquisa, a implementação de um controle P.I.D didático, realizado em Excel, onde é possível ver a aplicação prática de conceitos básicos de processamento digital de sinais.

6. REFERÊNCIAS

TEIXEIRA, E. P. **Notas de aula disciplina Controle de Processos** – Mestrado UNIUBE 2020.
NISE, N. S. **Engenharia de Sistemas de Controle**, 3ª Edição, LTC, 2002.
DORF, R. C & Bishop, R. H. **Sistemas de Controle Moderno**, Addison Wesley, 8ª Edição, 2001
OGATA, K. - **Engenharia de Controle Moderno**, Prentice-Hall, 4ª. ed., 2004.