IFAM - CMDI - ECAT

Controle Moderno

1 - Regras para Sintonia de Controladores PID

2 - Busca Numérica de Ganhos Otimizados

3 - Sintonia de Modelo Real

Flávio José Aguiar Soares - flaviosoares@ifam.edu.br

Julho, 2014

Capítulo 1

5^a aula

Regras para Sintonia de Controladores PID

Se a planta ou o sistema físico G(s) que se deseja controlar possuir um modelo matemático ou for possível se obter um, então os métodos analíticos de especificação de desempenho podem e devem ser aplicados.

Entretanto se G(s) for muito complexa ou desconhecida, então devem ser empregados técnicas experimentais para sintonizar os ganhos do controlador PID.

1.1 Aula de Laboratório - Exercícios sobre Sintonia de Controladores PID

Resolva os exercícios abaixo utilizando o programa Matlab.

1. Seja o sistema físico definido por G_p(s)= $\frac{2}{s^2+4}e^{(-s^2)}$

   a) Obtenha a resposta a uma entrada degrau para $\frac{2}{s^2+4}$;

   b) Obtenha a resposta a uma entrada degrau para $e^{(-s^2)}$;

   c) Apresente o polinômio correspondente a G_p;

   d) Obtenha a resposta a uma entrada degrau;

   e) O que você pode dizer com relação a estabilidade deste sistema?

   f) Este comportamento já era esperado? porquê?

   e) Considere o primeiro método de sintonia de Ziegler-Nichols. Isole a parte do gráfico similar a Figura 1.1. Estime os valores da constante de tempo T e do tempo morto L.

   $T =$

   $L =$

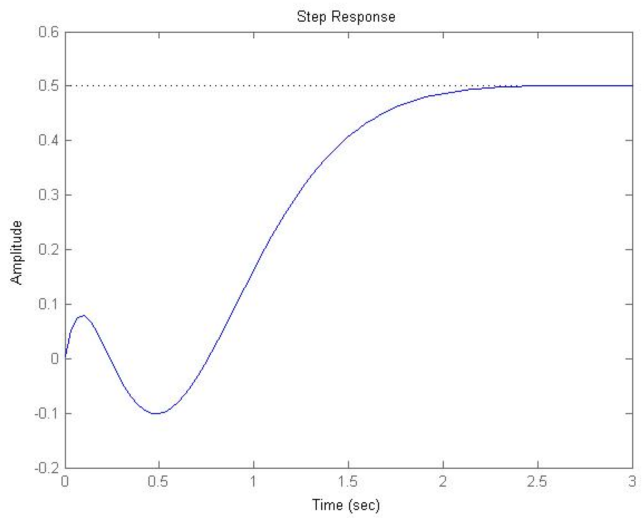

Figura 1.1: Transiente com atraso.

Tabela 1.1: Primeiro método de Ziegler - Nichols.

Controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	-	-
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	-
PID	$1,2\frac{T}{L}$	$2L$	$\frac{L}{2}$

f) Obter os ganhos para controladores PI usando o primeiro método de Ziegler-Nichols.

$P \Rightarrow K_p$=

$I \Rightarrow T_i$=

g) Obter os ganhos para controladores PID usando o primeiro método de Ziegler-Nichols.

$P \Rightarrow K_p =$

$I \Rightarrow T_i =$

$D \Rightarrow T_d =$

h) Obtenha a função de transferência do controlador $G_c(s)$, considerando os casos em que $G_c(s)$ seja um controlador PI ou um controlador PID. Primeiro método $Z - N$.

$PI \Rightarrow G_c(s) =$

$PID \Rightarrow G_c(s) =$

f) Obtenha a função de transferência do global $G(s)$, considerando os casos em que $G_c(s)$ seja um controlador PI ou um controlador PID. Primeiro método $Z - N$.

$PI \Rightarrow G(s) =$

$PID \Rightarrow G(s) =$

i) Considerando o segundo método de Ziegler-Nichols, obtenha o ganho crítico K_{cr} e o período crítico .

$K_{cr} =$

$P_{cr} =$

Tabela 1.2: Segundo método de Ziegler - Nichols.

Controlador	K_p	T_i	T_d
P	$\frac{K_{cr}}{2}$	-	-
PI	$0,45\frac{K_{cr}}{2}$	$\frac{P_{cr}}{1,2}$	-
PID	$0,6\frac{K_{cr}}{2}$	$\frac{P_{cr}}{2}$	$1,25P_{cr}$

j) Obter os ganhos para controladores PI usando o segundo método de Ziegler-Nichols.

$P \Rightarrow K_p =$

$I \Rightarrow T_i =$

l) Obter os ganhos para controladores PID usando o segundo método de Ziegler-Nichols.

$P \Rightarrow K_p =$

$I \Rightarrow T_i =$

$D \Rightarrow T_d =$

m) Obtenha a função de transferência do controlador $G_c(s)$, considerando os casos em que $G_c(s)$ seja um controlador PI ou um controlador PID. Segundo método $Z - N$

$PI \Rightarrow G_c(s) =$

$PID \Rightarrow G_c(s) =$

n) Obter a função de transferência global $G(s)$, considerando os casos em que $G_c(s)$ seja um controlador PI ou um controlador PID.Segundo método $Z - N$

$PI \Rightarrow G(s) =$

$PID \Rightarrow G(s) =$

o) Obter os ganhos para um controlador PI e para um controlador PID usando o método CHR para o problema servo e para o problema regulatótio com 20% de sobresinal.

$P \Rightarrow K_p =$

$I \Rightarrow T_i =$

$P \Rightarrow K_p =$

$I \Rightarrow T_i =$

$D \Rightarrow T_d =$

p) Obter os ganhos para controladores PID usando o método CC.

$P \Rightarrow K_p =$

$I \Rightarrow T_i =$

$D \Rightarrow T_d =$

q) Obter os ganhos para um controlador PI e PID usando o método da Integral do erro usando os critérios de desempenho IAE e ITAE.

$P \Rightarrow K_p =$

$I \Rightarrow T_i =$

$P \Rightarrow K_p =$

$I \Rightarrow T_i =$

$D \Rightarrow T_d =$

2. Refazer o problema anterior adotanto G_p(s)= $\frac{2s+25}{s^2+4s+25}$

1.1.1 Dicas para a resolução:

O Matlab opera polinômios na forma de vetores, então o formato da equação $G_p(s) = \frac{2}{s^2+4}e^{(-s^2)}$ precisa ser convertido em um poliômio equivalente. Isto pode ser feito utilizando a função *pade*. Apresentamos abaixo o help do Matlab para esta função:

PADE Pade approximation of time delays.

*[NUM,DEN] = PADE(T,N) returns the Nth-order Pade approximation of the continuous-time delay exp(-T*s) in transfer function form. The row vectors NUM and DEN contain the polynomial coefficients in descending powers of s.*

Veja que a função *pade* retorna um polinômio para a expressão $exp^{(-T*s)}$, de modo que o numerador e o denominador do polinômio que representam a função exponencial precisar ser multiplicado. pelo polinômino em s de $G_p(s)$.

A multiplicação de funções é denominada convolução. Para isso se usa a função *conv(função1,função2)* que irá tomar os dois argumentos e retornar os polinômiois da convolução.

Capítulo 2

6^a aula

Busca Numérica de Ganhos Otimizados

Definir o ganho do controlador PID é a tarefa que o projetista de controle deve ser capaz de resolver.

Existem diversos métodos experimentais, como vimos no capítulo anterior. Todos se baseiam no comportamento dinâmico da planta, ou seja na resposta de $G_p(s)$.

Veremos agora alguns métodos númericos que farão uma busca pelos ganhos ótimos para os elementos do controlador PID.

1 - **Busca Exaustiva:**

Nome técnico para dizer, chute! se não der chute de novo! Depois de algumas tentativas, com sorte você pode obter bons resultados. Mas este não é um método muito eficaz.

Neste método se procura uma solução que atenda as especificações de desempenho, ou seja os tempos de acomodação pelos critérios de 2% e 5%, máximo sobre-sinal, etc. Entretanto os ganhos obtidos não devem saturar os atuadores.

Se não sabe o que significa *saturar*, pergunte ao professor.

Quando o atuador satura se perde a ação de controle e a malha de realimentação deija de ter função ativa, ou seja, a malha de controle mesmo sendo fechada passa a se comportar como uma malha aberta.

A esta altura você já deve ter uma compreensão correta do que é malha fechada e malha aberta, se não tiver, pergunte ao professor.

2 - **Busca numérica:**

Neste método você define a faixa em que deseja que os seus ganhos estejam situados. Na prática isso é definido pela potência do seu atuador. Uma Ferrari faz de 0 a 100Km em pouco mais de 3s, em quantos segundos você acha que um fusca faz?

Evidentemente a faixa dos ganhos deve atender as especificações de desempenho. Estes valores são obtidos empregando-se um algorítmo de busca. A seguir veremos alguns exemplos retirado do Ogata que fazem esta busca.

Use o Matlab e reescreva estes exemplos em arquivos.m de modo a verificar o seu funcionamento. Você deve tentar compreender a função de cada linha de programção. O que ela faz e porque ela está ali. Ou seja, **comente cada linha de programação**.

2.0.1 Exemplos:

EX - 1: Dado o controlador $G_c(s) = k\frac{(s+a)^2}{s}$, se deseja determinar os valores de k e a para a FTMF de modo que dada uma entrada degrau unitário o máximo sobre-sinal seja $5\% < K < 10\%$.

Solução: Vamos supor inicialmente que $2 \leq M \leq 5$ e que $0,5 \leq a \leq 1,5$. Caso o algorítmo de busca não encontre uma solução os intervalos de busca devem ser ampliados e a busca reiniciada.

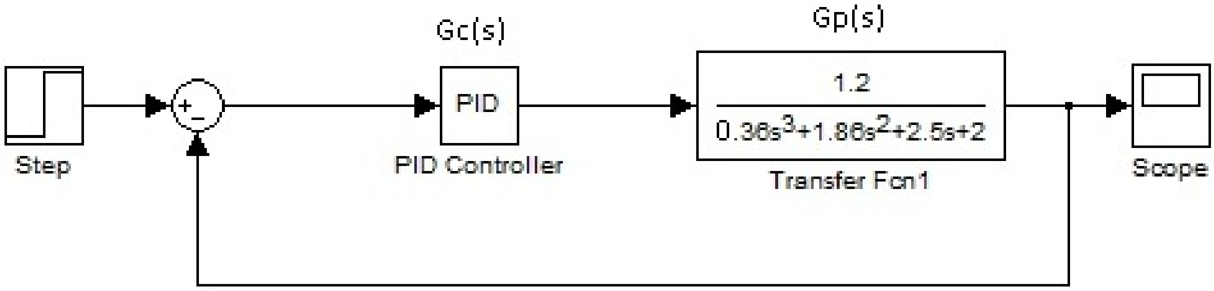

Figura 2.1: Malha de controle na qual se prentende obter os ganhos do PID.

Este algoritmo corresponde ao programa 10.2, página 566 e 568 do livro texto, 4ª edição.

Solução A:

Crie um arquivo.m no Matlab e copie o programa abaixo. tente entender qual a função de cada linha do programa e porque ela está ali. Se não entender... pergunte ao professor.

Procure escrever do lado da linha os seus comentários

Este algoritmo determinará a primeira resposta que satisfizer a condição 5

```
clear all
clc
t=0:0.01:8;
for k=5:-0.2:2;
   for a=1.5:-0.2:0.5;
      num=[0 0 1.2*k 2.4*k*a 1.2*k*a^2];
      den=[0.36 1.86 2.5+1.2*k 1+2.4*k*a 1.2*k*a^2];
      y=step(num,den,t);
      m=max(y);
      if m<1.1 & m>1.05
         break  %interrompe o loop interno que varia o parâmetro a
      end
   end
   if m<1.1 & m>1.05
       break % interrompe o loop externo que varia o parâmetro k
   end
end
plot(t,y)
grid
```

```
title('Resposta ao degrau unitário')
xlabel('t(s)')
ylabel('Saída')
ganhos=[k;a;m]
```

O algorítmo vai gerar um gráfico. Amplie a região do sobre-sinal e verifique se ele está dentro do intervalo de projeto desejado.

Observe que o algoritmo foi capaz de determinar os ganhos que fazem com que a resposta dinâmica esteja dentro do intervalo

Solução B:

Como exercício, *comente cada linha do algoritmo.*

Este algoritmo determinará a primeira resposta que satisfizer a condição 5

```
clear all
clc
t=0:0.01:8;
for k=5:-0.2:2;
   for a=1.5:-0.2:0.5;
      num1 = k*[1 2*a a^2];
      den1 = [0 1 0];
      tf1 = tf(num1,den1) % tf1 = Gc(s) = PID
      num2 = [0 0 0 1.2];
      den2 = [0.36 1.86 2.5 1];
      tf2 = tf(num2,den2) % tf2 = Gp(s)
      tf3 = tf1 * tf2     % tf3 = Gc(s)*Gp(s)
      sys = feedback(tf3,1);
```

Onde $sys = \frac{C(s)}{R(s)} = G(s) = \frac{G_c(s)*G_p(s)}{1+G_c(s)*G_p(s)}$

```
      y = step(sys,t)
      m = max(y)
      if m<1.1 & m>1.05
         plot(t,y);
         grid;
         title('Resposta do Sistema a um degrau unitário')
         xlabel('t(s)')
         ylabel('Saída')
         ganhos=[k;a;m];
         break  %interrompe o loop interno que varia o parâmetro a
      end
   end
   if m<1.1 & m>1.05
```

```
        break % interrompe o loop externo que varia o parâmetro k
    end
end
```

No exemplo anterior o algorítmo de busca para assim que encontra a primeira solução. Veremos a seguir um exemplo em que o algorítmo encontra um conjunto de possíveis soluções.

2.0.2 Lista de Exercícios:

Resolva os seguintes problemas propostos do livro texto, 4^a edição:

- Refaça os exemplos A.10.1, A.10.3, A.10.5, A.10.7, A.10.8, A.10.9.
- Faça os exercícios B.10.1, B.10.2, B.10.3, B.10.7.
- Refaça o exemplo A.10.11. Considere um controlador PID onde $G_c = k\frac{(s+a)^2}{a}$ que deve controlar a planta $G_p = \frac{1}{s(s+1)(s+5)}$. Obtenha:

  a) Um algorítmo que faça a busca automática dos ganhos. Determine o valores dos ganhos k e a de modo que o máximo sobre-sinal esteja dentro do intervalo $5\% < M < 10\%$ com tempo de acomodação $t_s < 3s$. Adote a região de busca $2 \leq k \leq 50$ e $0,05 \leq a \leq 2$.

  b) A função de transferência $G(s)$;

  c) A equação característica;

  d) Determine o máximo valor de ganho para que a resposta a uma entrada degrau seja estável.

  Lembre-se que existem algumas maneiras de verificar a estabilidade de uma função, pelo método de Routh ou pela posição dos pólos da equação característica. Os pólos de uma função correspondem as suas raízes. As raízes de um polinômio pode ser obtido utilizando a função *roots()*.
- B.10.11 utilizando o Matlab.
- B.10.12 utilizando o Matlab.

Capítulo 3

7^a aula - Trabalho Individual - Sintonia de um Modelo Real

Considere o modelo de um motor DC, controlado pela armadura, figura (3.1), cujos dados reais estão dispostos na tabela (3.1).

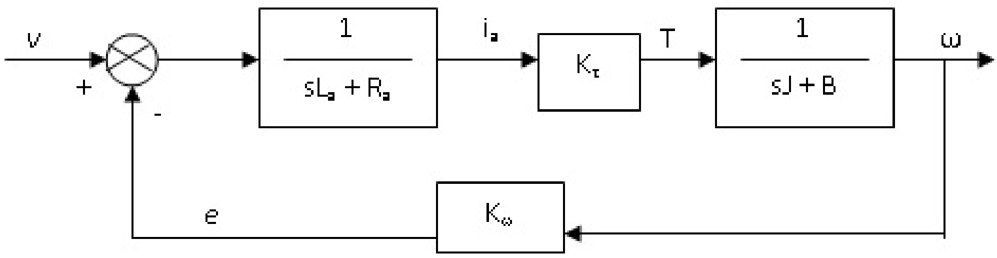

Figura 3.1: Modelo de blocos de um motor DC de ímã permanente.

Tabela 3.1: Parâmetros reais de um motor DC.

Descrição	**Símbolo**	**Valor**
Resistência da bobina da armadura	R_a	7,7 Ω
Indutância da bobina da armadura	L_a	4,92 mH
Momento de inércia	J	$5X10^{-3}\frac{Nm^2}{rad}$
Atrito viscoso	B	$100\mu\frac{Nms}{rad}$
Constante de torque	K_t	$5X10^{-4}$
Constante de velocidade	K_w	$5X10^{-4}$
Ganho do tacômetro	H_g	$1\ \frac{V}{rad}$

Utilize um controlador PI conforme a figura(3.2).

1. Refaça o modelo da figura(3.2) no simulink para auxiliar sua compreensão;

2. Escolha e aplique um dos métodos de sintonia estudados e obtenha os ganhos;

3. Aplique uma entrada degrau e verifique a resposta;

4. Escreva um algoritmo que encontre os ganhos;

5. Reaplique estes ganhos ao modelo para uma entrada degrau unitário;

6. Utilize a opção de autosintonia do bloco PID do *simulink*, ajuste o tempo de resposta e verifique a curva de saída do sistema controlado;

7. Compare, analise e comente os resultados.

8. Apresente um relatório que descreva todos os procedimentos anteriores. Inclua os gráficos seja do Matlab, seja do Simulink.

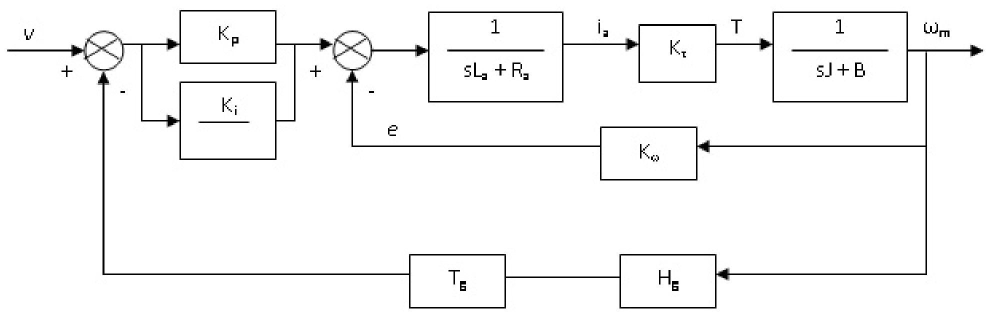

Figura 3.2: Controlador PI aplicado ao modelo do motor DC.

Repita o procedimento anterior, agora para um controlador PID, conforme a figura (3.3) abaixo,

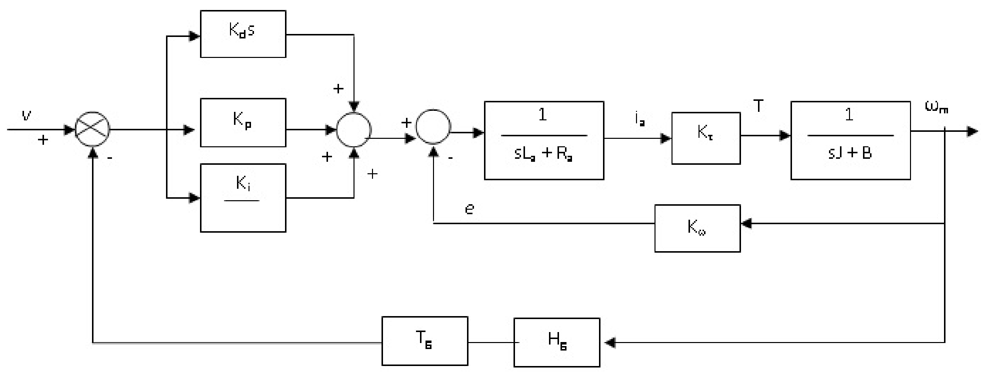

Figura 3.3: Controlador PI aplicado ao modelo do motor DC.