

AVALIAÇÃO 2 - CONTROLE 1

Aluno: Vitor de Sousa França
Matrícula: 20180041455
27 de Outubro de 2021

Problemas PID:

Considerando-se o sistema:

$$G = \frac{0.25(K_d s^2 + K_p s + K_i)}{s(s+1)(s+5)}$$

Em que $K_d = 150.88$, $K_p = 1373.92$ e $K_i = 5000$

a) Utilize o MATLAB e realize as simulações do sistema em malha fechada no domínio do tempo contínuo.

Primeiro, foram definidas as funções de transferência (FT) da planta, do controlador e então obteve-se o ganho em malha aberta (MA) e em malha fechada (MF), posto que esse conjunto de FT serão utilizados por várias questões. O bloco de código 1 apresenta

Código 1: Definindo FTs

```
1 %Planta
2 Nump = 0.25;
3 Denp = conv([1 1], [1 5]);
4 H = tf(Nump, Denp);
5
6 %Controlador
7 Kp = 1373.92;
8 Ki = 5e3;
9 Kd = 150.88;
10
11 Numc = [Kd Kp Ki];
12 Denc = [1 0];
13 Gc = tf(Numc, Denc);
14
15 %Ganho em Malha Aberta
16 Gma = H*Gc;
17
18 %Ganho em Malha Fechada
19 Gmf = Gma/(1+Gma);
```

Para observar o comportamento do sistema em MF, utilizou-se da entrada a degrau, como mostrado no bloco de código 2.

Código 2: Resposta ao degrau

```
1 %% Q a)
2 [y,t] = step(Gmf, 0.35);
3 figure
4 plot(t, y, 'LineWidth', 2);
5 legend('c(t)')
6 grid
```

Ao utilizar `[y,t] = step(Gmf, 0.35);` a função `step` não gera um gráfico mas sim, retorna vetores “y” e “t” referentes às posição dos pontos. Essa abordagem foi utilizada nesse e nos próximos códigos com a finalidade de realizar ajustes no gráfico como o de adicionar legenda e o de aumentar a espessura da curva. O resultado obtido pode ser visto na Figura 1.

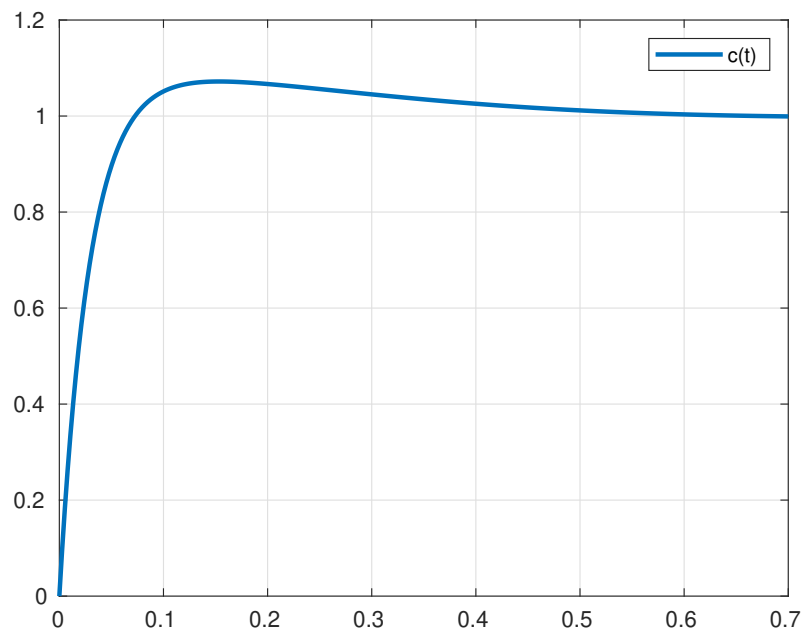


Figura 1: Gráfico da resposta ao degrau do sistema

b) Determine o LGR sem e com controlador

O lugar geométrico das raízes (LGR) pode ser facilmente obtido através da função `rlocus` nativa do matlab. O sistema sem o controlador é constituído unicamente da planta do sistema. O LGR do sistema com o controlador, por outro lado, é o produto no domínio “s”, da FT do controlador com a FT da planta. O bloco de código 3, apresenta o script desenvolvido para obtenção dos gráficos.

Código 3: LGRs do sistema

```
1 %% Q b)
2 %LGR sem controlador
3 figure
4 rlocus(H)
5 title('LGR sem controlador')
6 axis([-6 0.1 -5 5])
7 grid
8 %LGR com controlador
9 figure
10 rlocus(Gma)
11 axis([-6 0.1 -5 5])
12 title('LGR com controlador')
13 grid
```

A Figura 2 apresenta o LGR do sistema sem o controlador e a Figura 3 apresenta o LGR do sistema com o controlador.

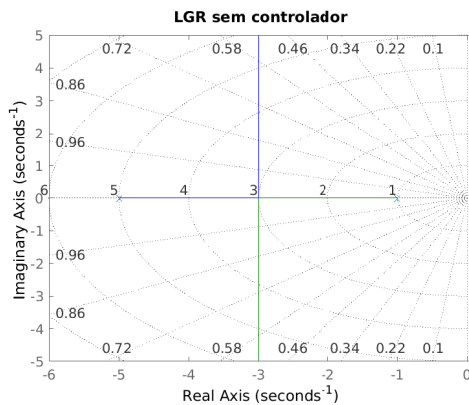


Figura 2: LGR do sistema sem o controlador

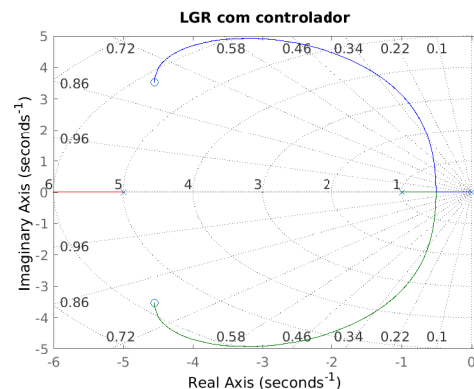


Figura 3: LGR do sistema com o controlador

c) Determine o tempo de amostragem.

O tempo de amostragem pode ser determinado de diferentes formas, desde que seja suficiente para não ocasionar problemas de *aliasing*, ou seja, sobreposição dos espectros. Para isso, sabe-se pelo Teorema de Nyquist que o período de amostragem “ T_s ” deve ser tal que a frequência de amostragem “ f_s ” seja o dobro da frequência de corte “ f_c ” do sistema, $\frac{1}{T_s} = f_s \geq f_c$.

O Teorema de Nyquist é bem fundamentado e pode ser amplamente utilizado, porém outra abordagem é amplamente utilizada no ambiente prático, dado a não trivialidade da obtenção da frequência de corte do sistema. Sob essa abordagem é feita uma aproximação de que o tempo de amostragem deve ser menor ou

igual ao tempo de subida da resposta ao degrau do sistema “ T_r ”, dividido por dez: $T_s = \frac{T_r}{10}$. Essa será a abordagem utilizada para todas as questões.

O bloco de código 4, apresenta a obtenção do tempo de amostragem. Nesse primeiro momento, verificou-se além do tempo de subida, a frequência de corte do sistema, para corroborar o fato de que o Teorema de Nyquist está sendo cumprido.

Código 4: Tempo de amostragem

```
1 %% Q c)
2 figure
3 bode(Gmf)
4 set(findall(gcf,'type','line'), 'linewidth', 2)
5 grid
6 % Tempo de amostragem
7 Ts = stepinfo(Gmf).RiseTime/10;
```

O tempo de amostragem é um atributo de nome `RiseTime` do método nativo `stepinfo(Gmf)`, dessa forma, pode-se obtê-lo utilizando `stepinfo(Gmf).RiseTime`. O tempo de subida da resposta ao degrau foi de $T_r = 48$ ms, então, o tempo de amostragem foi $T_s = 4,8$ ms e a frequência de amostragem $f_s = 209.1734$ Hz. Ao verificar o diagrama de bode, Figura 4, será possível determinar a frequência de corte do sistema.

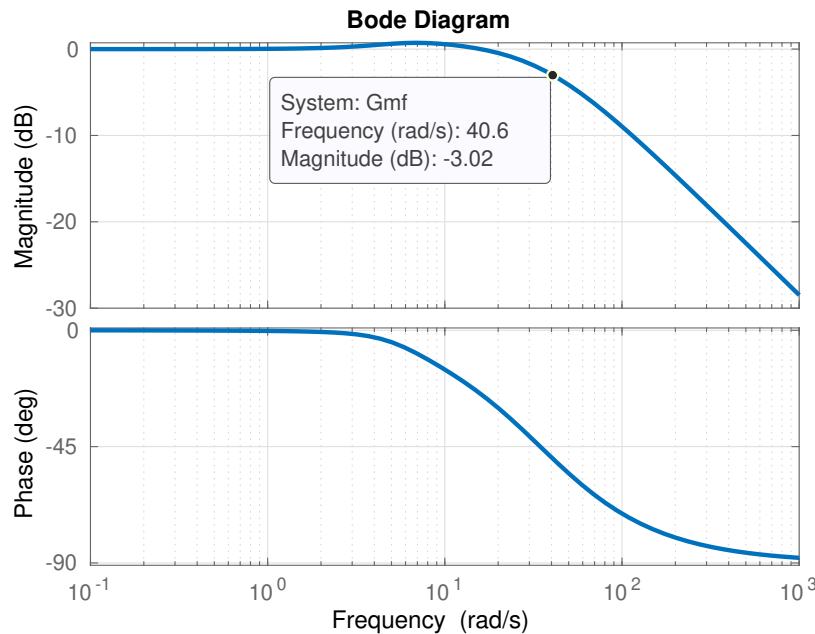


Figura 4: Diagrama de Bode

A frequência de corte é aquela no qual o ganho do sistema é de -3 dB. Pelo gráfico, o ponto está aproximadamente na frequência de corte do sistema $f_c = \frac{\omega_c}{2\pi} = \frac{40.6}{2\pi} = 6.4617$ Hz.

Podemos verificar que a técnica amplamente utilizada em ambientes práticos nos retorna frequências de amostragens muito maiores que a frequência de corte e garantem assim que a discretização esteja satisfazendo ao Teorema de Nyquist. De toda forma, é interessante perceber que ao utilizar a aproximação dada por esse método o nosso resultado não será o mais otimizado e poderá pecar em termos de custo de projeto.

d) Discretize o controlador.

O controlador PID, é um controlador clássico que no domínio “s” de Laplace, é definido pela FT apresentada na equação 1

$$G_c(s) = kp + \frac{k_i}{s} + k_d s \quad (1)$$

Para discretizar o controlador, devemos passar a FT do domínio “s” para o domínio discreto ‘z’, seguindo a correspondência $s = \frac{1}{T_s} \ln(z)$, em quem “ T_s ” é o período de amostragem. Porém, visto que iremos obter resultados não lineares, essa transformação poderá ser custosa computacionalmente. Então, frequentemente são utilizadas aproximações para essa equação. Nessa avaliação será utilizado o método de Tustin, também conhecido como aproximação bilinear ou trapezoidal, apresentada pela equação 2.

$$s = \frac{T_s}{2} \cdot \frac{1 + z^{-1}}{1 - z^{-1}} \quad (2)$$

Realizando a substituição da equação 2 em 1, obtemos a equação 3.

$$G_c(z) = \frac{(k_p + \frac{T_s k_i}{2} + \frac{2k_d}{T_s})z^2 + (T_s k_i - \frac{4K_d}{T_s})z - k_p + \frac{T_s k_i}{2} + \frac{2k_d}{T_s}}{z^2 - 1} \quad (3)$$

Ao substituir os valores do tempo de amostragem e dos ganhos k_p , k_i , k_d , na equação 3 obtem-se a FT discreta 4.

$$G_c(z) = \frac{(6,451 \cdot 10^4)z^2 - (1,262 \cdot 10^5)z - 6,176 \cdot 10^4}{z^2 - 1} \quad (4)$$

A discretização também pode ser realizada através de uma função `c2d` nativa do Matlab. Utilizando `c2d(Gc, Ts, 'tustin')` em que `Gc` foi a função de transferência do controlador no domínio “s” e `Ts` o período de amostragem, obteve-se a mesma função de transferência 4.

e) Realize simulações do sistema em malha fechada e faça uma comparação entre a resposta no tempo contínuo e discreto.

A simulação realizada é apresentada através do bloco de código 5 em que o sistema é discretizado e verifica-se sua resposta a uma entrada degrau. Com a finalidade de comparação, foi realizado um gráfico das respostas contínua e discretizada sobrepostas que pode ser visto na Figura 5.

Código 5: Simulações do Sistema em MF e em MA.

```
1
2 Gmfz = c2d(Gmf, Ts, 'zoh');
3 [yz,tz] = step(Gmfz);
4 figure
5 stairs(tz, yz, 'LineWidth', 2);
6 legend('c(kT)')
7 grid
8 %% Q f)
9 figure
10 plot(t, y, 'LineWidth', 2)
11 hold on
12 stairs(tz, yz, 'LineWidth', 2);
13 hold off
14 legend('c(t)', 'c(kT)')
15 grid
```

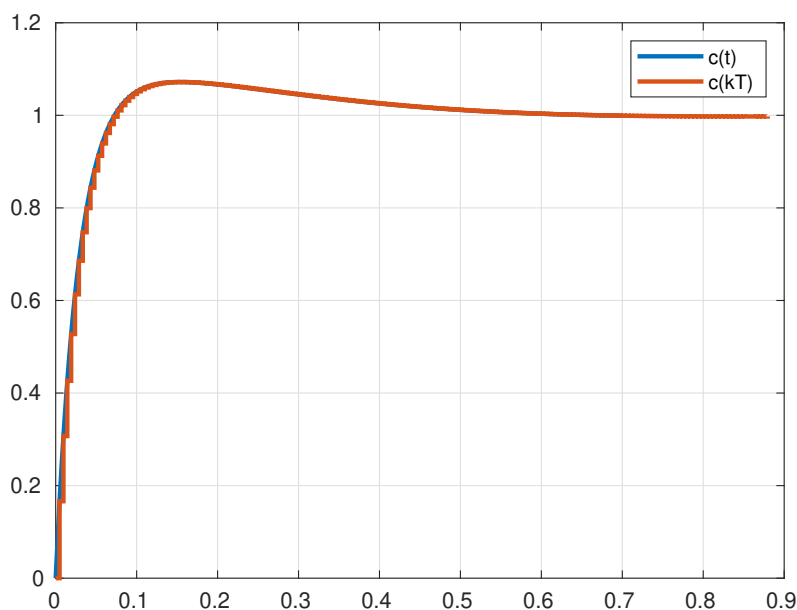


Figura 5: Gráfico das respostas ao degrau dos sistemas contínuo e discreto

Além da resposta gráfica, faz-se interessante avaliar numericamente o erro para que seja possível verificar a qualidade da aproximação. Para isso, construiu-se um *script* capaz de calcular o erro absoluto médio percentual (MAPE), que pode ser visto no bloco de código 6.

Código 6: Erro Absoluto Médio Percentual

```
1 ape = abs((yz - y(1:length(yz)))/y(1:length(yz)));
2 mape = mean(ape(isfinite(ape))); %retira o erro percentual do y=0
```

Algumas considerações devem ser feitas para compreender o bloco de código 6. Pela alteração do tempo de amostragem da função step o tamanho dos vetores da saída do sistema em tempo contínuo e em tempo discreto diferem. Portanto, foi necessário realizar um *slice* do maior vetor para que assim fosse possível realizar as operações necessárias.

Junto a isso, como a saída do sistema em $t = 0s$ é $y(0) = 0$, a divisão pelo vetor de saídas gera um elemento que tende ao infinito, para retirá-lo só é realizada a média dos valores finitos através da função `isfinite` que retorna um vetor de booleanos, em que “1’s” estarão na posição do vetor que são finitos e “0’s” na posição dos infinitos. Ao fim, obteve-se um $MAPE = 0.024\%$, um valor muito baixo de erro.

f) Realize a análise de estabilidade do sistema usando o método de Jury.

O método de Jury permite avaliar a estabilidade de sistemas discretos ao analisar o polinômio característico. Existem alguns critérios que devem ser satisfeitos para que o sistema seja estável. Os critérios são:

$$Q(1) > 0 \quad (1)$$

$$(-1)^n Q(-1) > 0 \quad (2)$$

$$|a_0| < a_n \quad (3)$$

$$|b_0| > b_{n-1} \quad (4)$$

$$|c_0| > c_{n-2} \quad (5)$$

$$\vdots \quad \vdots$$

O polinômio característico do sistema que está sendo analisado pode ser visto na equação 5.

$$Q(z) = z^3 - 2.804z^2 + 2.616z - 0.8114 \quad (5)$$

A quantidade de critérios é dada pelo grau do polinômio mais um. Para o nosso caso, em que o polinômio característico é de terceiro grau, são portanto, quatro critérios.

- O primeiro critério: $Q(1) = 6e - 4 > 0$, satisfeito;
- O segundo critério: $-1^3 Q(-1) = 5.2314 > 0$, também trata-se de uma verdade;
- O terceiro critério: $0.8114 < 1$, satisfeito.

Os próximos critérios só são possíveis de ser avaliados através da Tabela 1. Para nosso caso só é necessário verificar mais um critério.

Tabela 1: Análise de estabilidade

	z^0	z^1	z^2	z^3
1	-0,8114	2,616	-2,804	1
2	1	-2,804	2,616	-0,8114
3	-0,3408	0,682	-0,3416	0

- O quarto critério: $0.3416 > 0.3408$, satisfeito.

Problemas PI:

Para os problemas apresentados, realizar:

- Simulação usando MATLAB/Simulink realizando análise dos resultados com relação a estabilidade e resposta dinâmica do sistema
- Analisar o LGR sem/com controlador
- Determine o tempo de amostragem e realize a discretização dos controladores. Comparar com a solução contínua.
- Discretizar os sistemas (controlador e planta) e analisar a estabilidade usando o método de Jury e Routh Hurwitz.

Problema 1:

Para esse problema, o modelo de um motor possui como FT a equação 6.

$$G_p = \frac{4}{s^3 + 3s^2 + 10s} \quad (6)$$

A FT de MF, com um controlador proporcional é o visto na equação 7.

$$Gmf = \frac{4kp}{s^3 + 3s^2 + 10s + 4kp} \quad (7)$$

Foi visto no desenvolvimento da questão que para o sistema ser estável o ganho proporcional deve estar no intervalo: $0 < kp < 7,5$. Foi determinado $kp = 4$.

a)

A estabilidade do sistema foi verificada através de uma entrada ao degrau. Pelo desenvolvimento analítico, o erro em regime estacionário deve ser zero a uma entrada ao degrau, espera-se obter o mesmo resultado pela simulação no MATLAB através do Código 7.

Código 7: Análise da estabilidade

```
1 %FT sem controlador
2 num = 4;
3 den = [1 3 10 0];
4 G = tf(num,den);
5 %FT com controlador
6 kp = 4;
7 Gmf = tf(kp*num,[1 3 8 4*kp]);
8
9 % a)
10 [y,t] = step(Gmf);
11 figure
12 plot(t, y, 'LineWidth', 2);
13 legend('c(t)')
14 grid
```

A Figura 6 apresenta a resposta do sistema ao degrau.

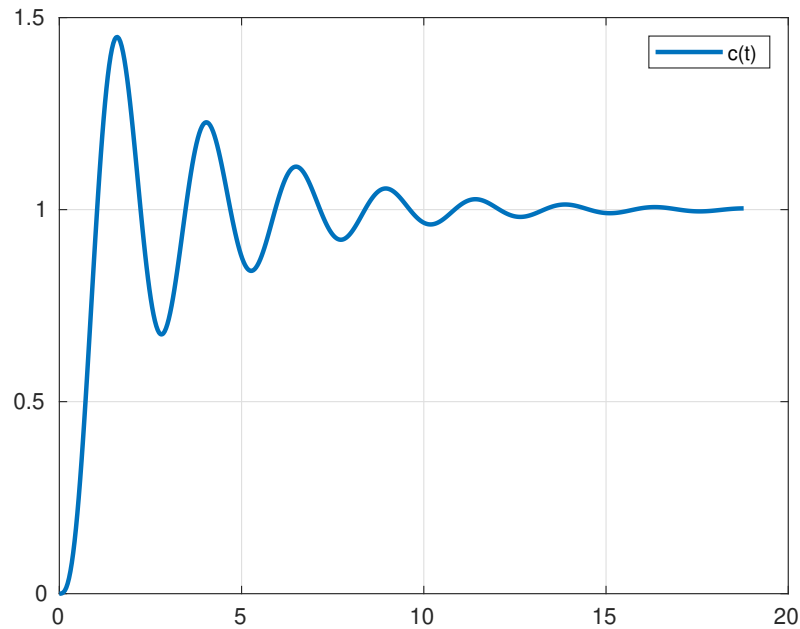


Figura 6: Gráfico da resposta ao degrau do sistema

Podemos verificar que o erro em regime estacionário do sistema tende a zero, assim como determinado através do cálculo analítico.

b)