Lista 1 - PTC-5719 Identificação de Sistemas

Mateus Chiqueto dos Santos

Abril 2025

1 a.

1.1 Resolução

Inicializamos as variáveis 'K', 'tau', 'theta', e também o numerador e denominador da função contínua. Para a discretização é utilizada a função c2d e 'zoh' indica que a função irá utilizar o método de segurador de ordem zero para a discretização. Segue abaixo o código da discretização e do processo para simulação da função discreta e contínua.

Listing 1: Código MATLAB para simulação do sistema contínuo e discreto

```
K = 3;
tau = 10;
theta = 5;
num = K;
den = [tau 1];
G = tf(num, den, 'InputDelay', theta);
T = 1;
Gd = c2d(G, T, 'zoh');
```

Segue a equação discretizada:

$$G_d = \frac{0.2855}{z - 0.9048}.$$

Para a simulação utilizaremos a função lsim do matlab e é necessário um vetor de tempo e entrada de degrau, em que a função lsim irá utilizar para realizar a simulação.

Listing 2: Código MATLAB para simulação do sistema contínuo e discreto

```
t = 0:0.01:20;
u = 0.1 * ones(size(t)); % Degrau de 0,1
[y_cont, t_cont] = lsim(G, u, t);

t_disc = 0:T:20;
u_disc = 0.1 * ones(size(t_disc));
[y_disc, t_disc_out] = lsim(Gd, u_disc, t_disc);
```

Agora vamos plotar o gráfico com as duas simulações. A simulação discreta utiliza do pacote 'stairs' e continua do 'plot'.

Listing 3: Código MATLAB para simulação do sistema contínuo e discreto

```
figure;
plot(t_cont, y_cont, 'b', 'LineWidth', 2); hold on;
stairs(t_disc, y_disc, 'r', 'LineWidth', 2);
xlabel('Tempo (s)');
ylabel('Saida');
legend('Continuo', 'Discreto');
title('Sistema Continuo e Discreto');
grid on;
```

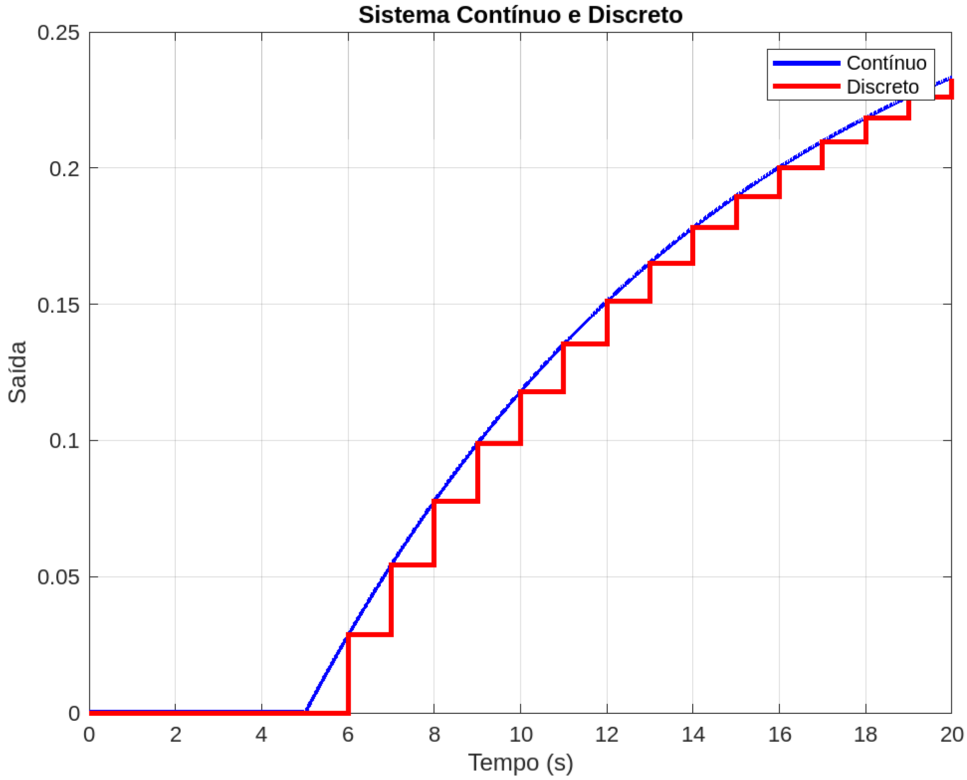


Figure 1: Comparação das respostas do sistema contínuo e discreto

1.2 Conclusão

As duas simulações sempre se encontram ao fim de 1 segundo, porém entre um segundo e outro existe um atraso da simulação discreta em relação a simulação contínua, porém esse atraso se diminui ao longo da simulação, visualmente, nota-se que o atraso diminui cerca de metade a cada segundo de simulação. Portanto, na fase inicial as simulações podem obter resultados em grande diferença.

2 b.

Para este problema, iremos inicializar o ruído no MATLAB e somar sua sáida com a saída do sistema contínuo simulado anteriormente e plotar sua simulação.

A equação $ruido = media + desviopadrao * randn(size(y_{cont}))$ gera exatamente o ruído necessário para a simulação. A função do randn do MATLAB gera números aleatórios na mesma dimensão das saídas da simulação contínua.

Segue o código em MATLAB.

Listing 4: Código MATLAB para simulação do sistema contínuo com ruido gaussiano.

```
media = 0;
variancia = 1e-6;
desvio_padrao = sqrt(variancia);
ruido = media + desvio_padrao * randn(size(y_cont));
y_cont_com_ruido = y_cont + ruido;

figure;
plot(t_cont, y_cont_com_ruido, 'b', 'LineWidth', 1);
xlabel('Tempo (s)');
ylabel('Saida com Ruido');
title('Sistema Continuo com Ruido');
grid on;
```

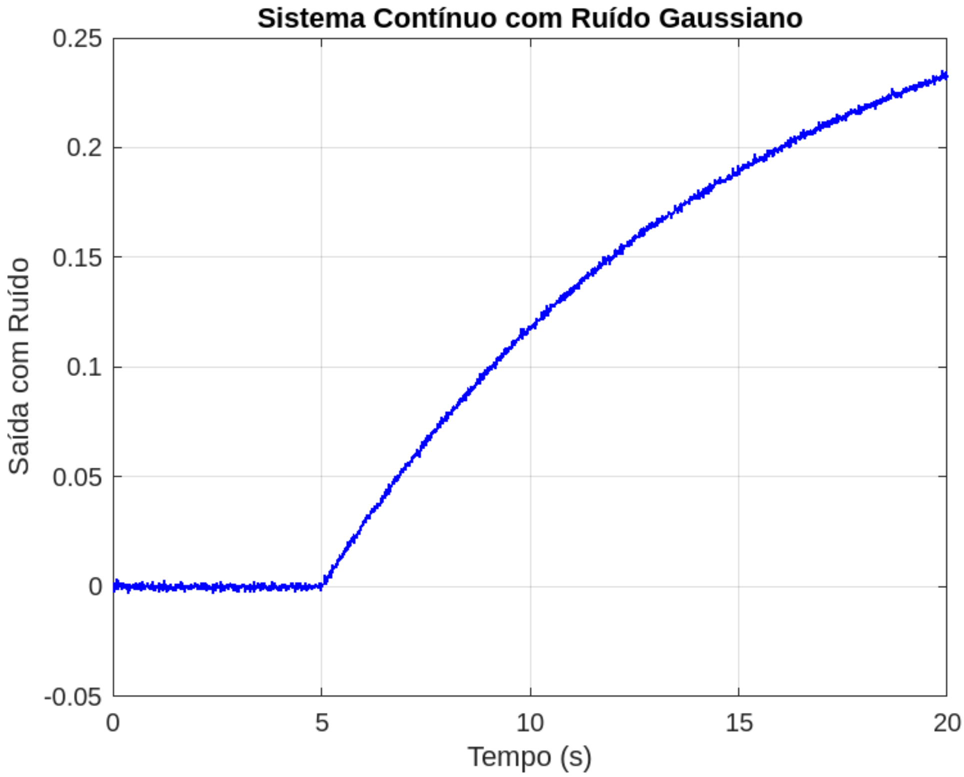


Figure 2: Simulação do sistema contínuo com ruído gaussiano

3 c.

3.1 Conclusão sobre as perturbações.

É possível notar que a perturbação direta tanto com alta variancia tanto com baixa tem alta variação dos sinais. Com alta variação não se observa a curva de resposta sem ruído mostrada anteriormente. Com filtro de primeiro grau podemos notar que ocorre uma suavização dos ruídos encontrados sem filtro, as grandes variações são removidas, principalmente quando em baixa variancia. Já com filtro de segunda ordem a resposta é suave e poucos ruídos são notados.

3.2 Discretização das duas perturbações.

Do mesmo modo que realizamos a discretização no primeiro exercício, realizaremos agora, veja o código em MATLAB.

Listing 5: Código MATLAB para discretização das perturbações.

```
G_pert1 = tf(1, [5 1]);
G_pert1_d = c2d(G_pert1, 1, 'zoh');

G_pert2 = tf(2, [50 15 1]);
G_pert2_d = c2d(G_pert2, 1, 'zoh');
```

As perturbações discretizadas são:

$$G_{\text{pert1}_d}(z) = \frac{0.1813}{z - 0.8187},$$

e

$$G_{\text{pert2}_d}(z) = \frac{0.01811z + 0.01639}{z^2 - 1.724z + 0.7408}.$$

4 d.

Os gráficos das perturbações são encontrados nas figuras 7, 8, 3, 4, 5 e 6. Após análise, é possível notar que a perturbação que mais afeta a saída do sistema é a perturbação direta, sem filtro, e ainda com intensidade de alta variancia. E também, neste caso ainda os filtros de primeira e segunda ordem suavizam a perturbação. Porém, quando a intensidade é alta, os filtros seguem os ruídos retirando altas variações, porém os resultados são relevantemente modificados, onde é possível notar na figura 3 e na figura 2.

5 e.

Nas figuras 10 e 9, vemos, respectivamente, os gráficos das simulações com amplitude alta e amplitude baixa.

6 f.

É possível realizar a análise de correlação com a curva do processo com perturbação de baixa intensidade. Nesse caso podemos estimar um modelo de primeira ordem com atraso, agora precisaremos estimar τ, θ e K.

Para K, faremos a razão entre variação da saída em regime permanente e amplitude do degrau. Será:

$$K_est = (y_{final} - y_{inicial})/0.1.$$

Já para τ e θ faremos:

$$tau_est = 0.67 * (t_{85.3} - t_{35.3});$$

$$theta_est = 1.3 * t_{35.3} - 0.29 * t_{85.3},$$

tais que $t_{85.3}$ é o momento em que a resposta da simulação atinge 85.3% do valor final, semelhante para $t_{35.3}$. Na figura 11 podemos observar a curva do processo original em azul e do modelo estimado em vermelho.

O tempo de acomodação será medido pela função "*info.SettlingTime*" do MATLAB e após a simulação temos que $T_s = 41.1225s$.

Listing 6: Código MATLAB para simulação do modelo estimado e comparação com o sistema.

```
>>
tempo = out.tout;  % Vetor de tempo
saida = out.simout;
>>
y_inicial = saida(1);
y_final = mean(saida(end-10:end));
K = (y_final - y_inicial)/0.1;  % ganho (degrau de 0.1)

y_35 = y_inicial + 0.353*(y_final - y_inicial);
y_85 = y_inicial + 0.853*(y_final - y_inicial);

idx_35 = find(saida >= y_35, 1);
idx_85 = find(saida >= y_85, 1);
t_35 = tempo(idx_35);
t_85 = tempo(idx_85);

tau = 0.67*(t_85 - t_35);
theta = 1.3*t_35 - 0.29*t_85;
>>
num = K;
den = [tau 1];
G_est = tf(num, den, 'InputDelay', theta);
>>
t_sim = 0:0.1:60;
```

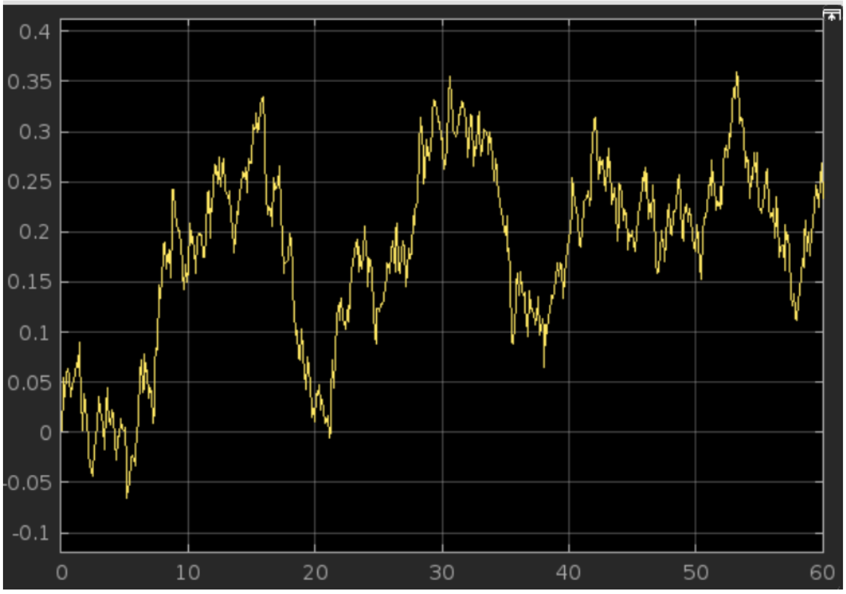


Figure 3: 1 ordem alta

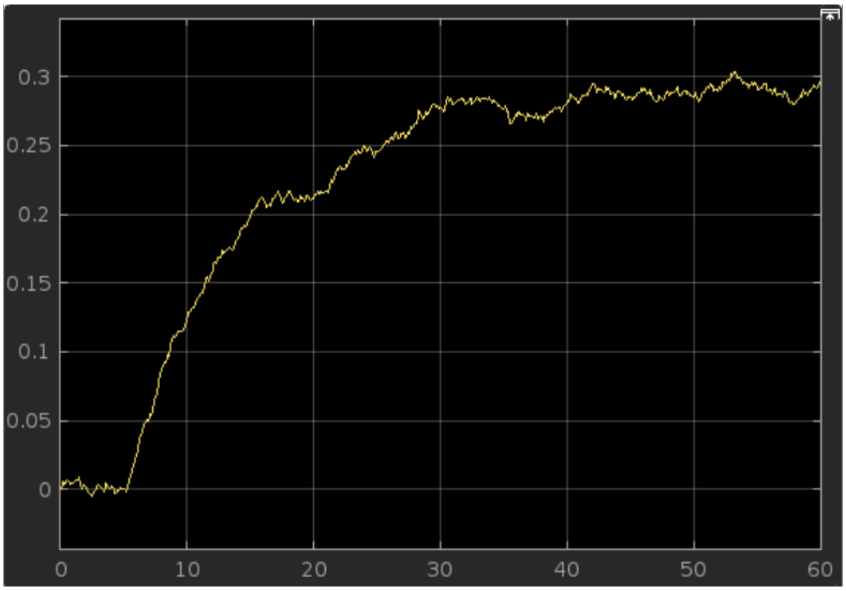
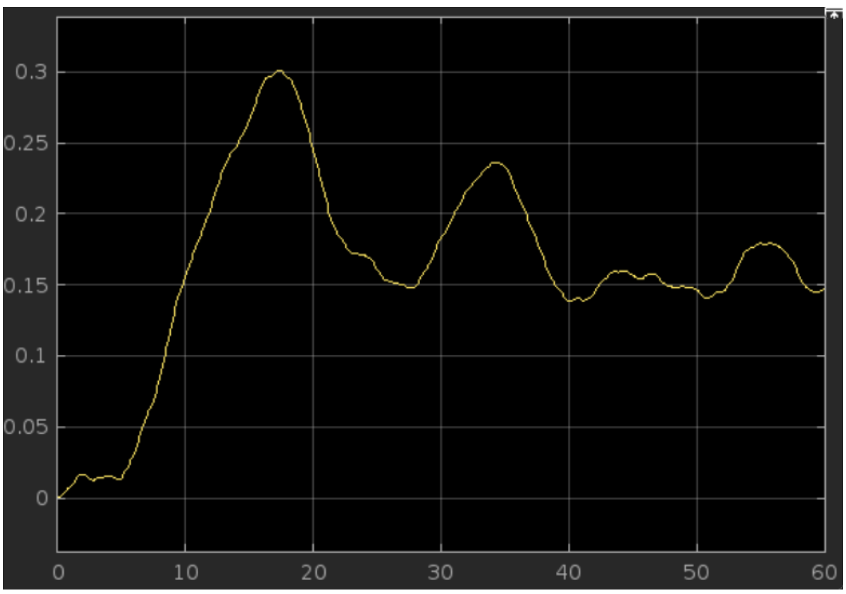


Figure 4: 1 ordem baixa



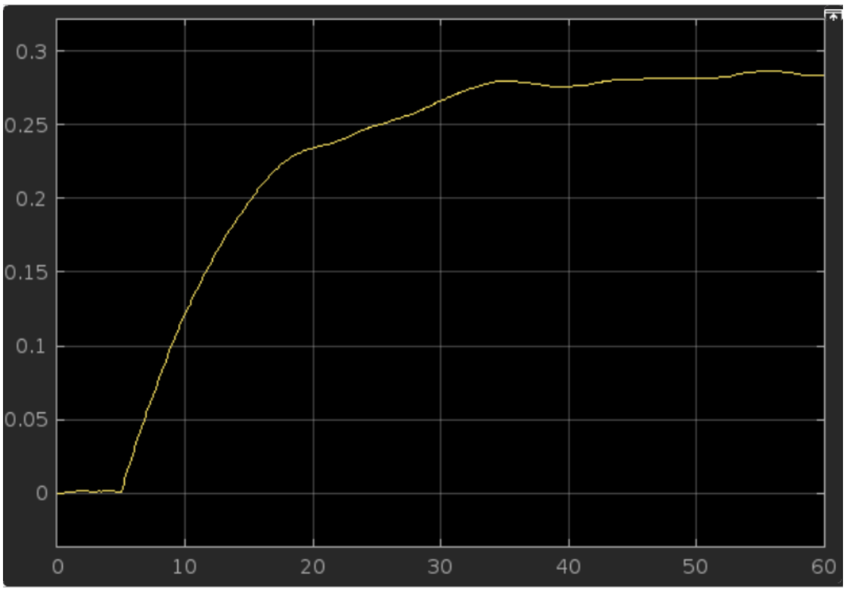


Figure 6: 2 ordem baixa

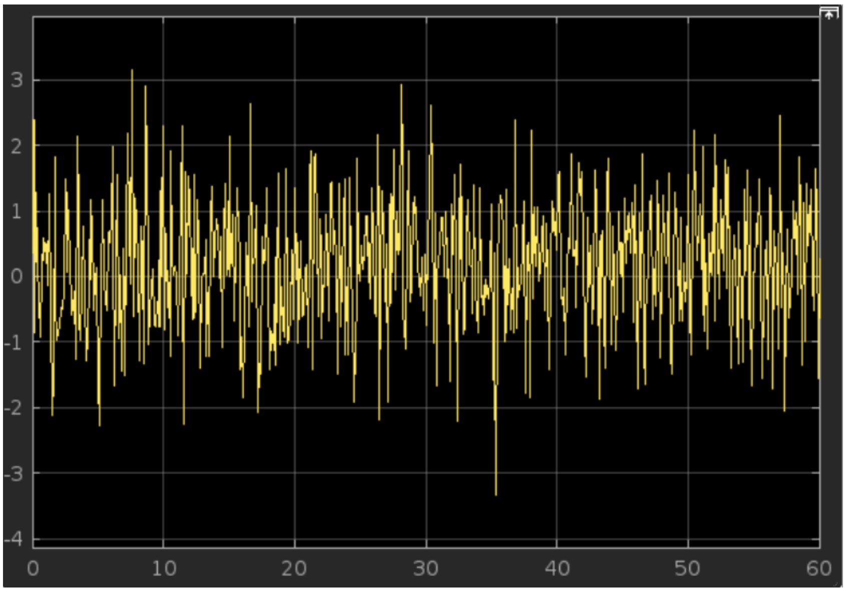
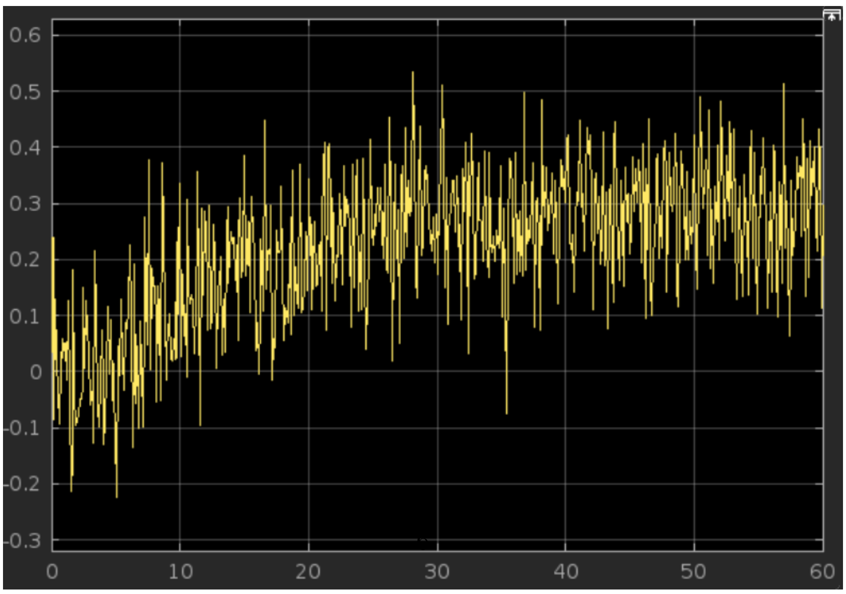


Figure 7: direta alta



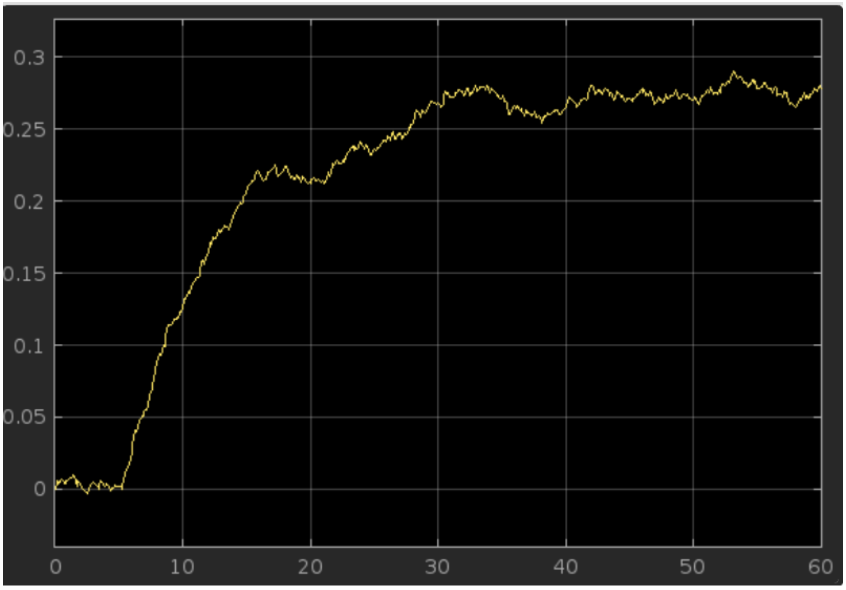


Figure 9: Amplitude baixa

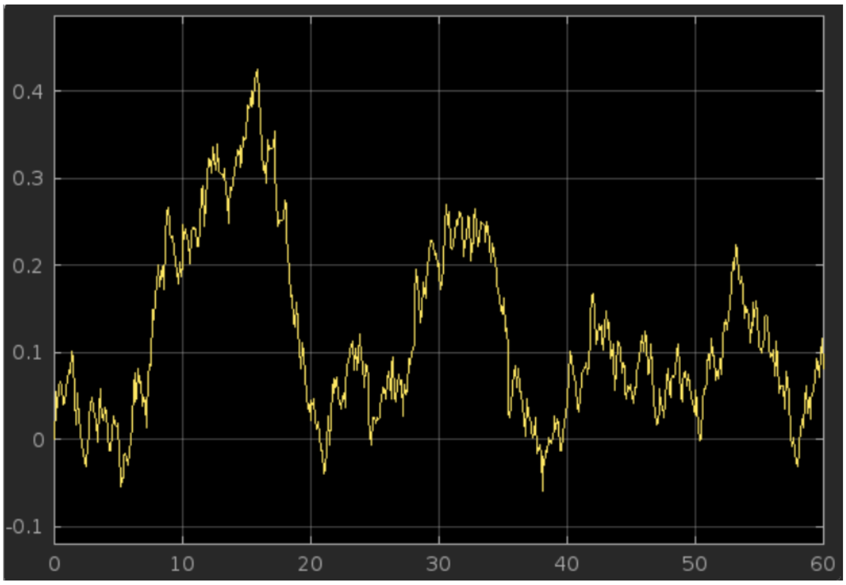


Figure 10: Amplitude alta

```
u = 0.1 * ones(size(t_sim));
[y_est, t_est] = lsim(G_est, u, t_sim);
>>
num = 3;
den = [10 1];
G_org = tf(num, den, 'InputDelay', 5);
t_org = 0:0.1:60;
```

Listing 7: Código MATLAB para calculo do tempo de acomodação.

```
[y_est, t_sim] = step(G_est, 60);
info = stepinfo(y_est, t_sim);

settlingTime = info.SettlingTime

settlingTime =

    41.1225
```

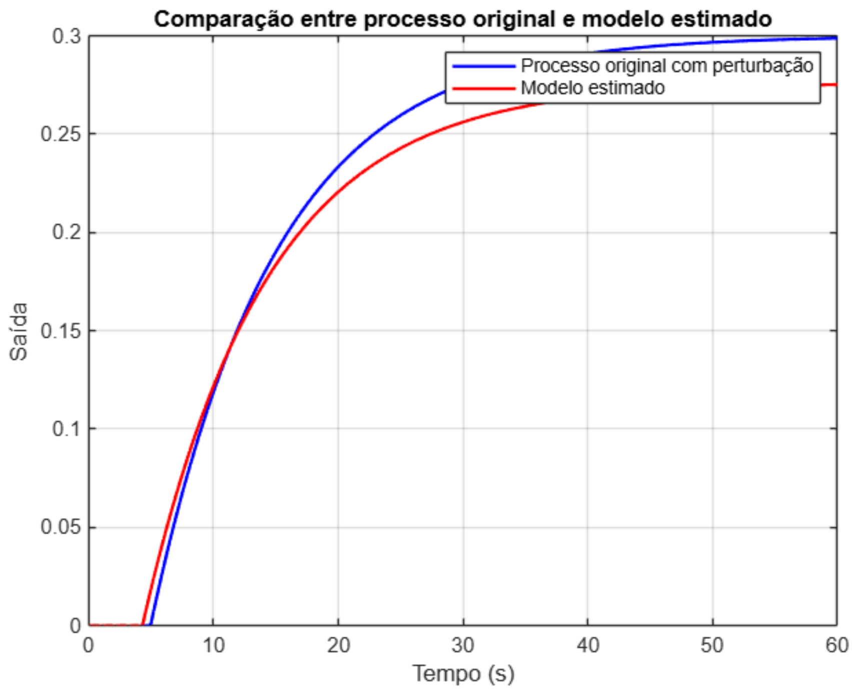


Figure 11: Comparação entre simulação real e a simulação estimada

7 g.

Nesse caso simulamos por $2T_s = 82s$ e com pulso unitário. Na resposta da simulação com perturbações de baixa intensidade é possível observar o comportamento da função-peso, que nesse caso é a simulação limpa, sem perturbações.

8 h.

Dado o seguinte código de MATLAB foi possível obter as curvas da convolução entre pulso degrau e a resposta da questão anterior. Na figura **??** podemos observar as curvas geradas pelas convoluções.

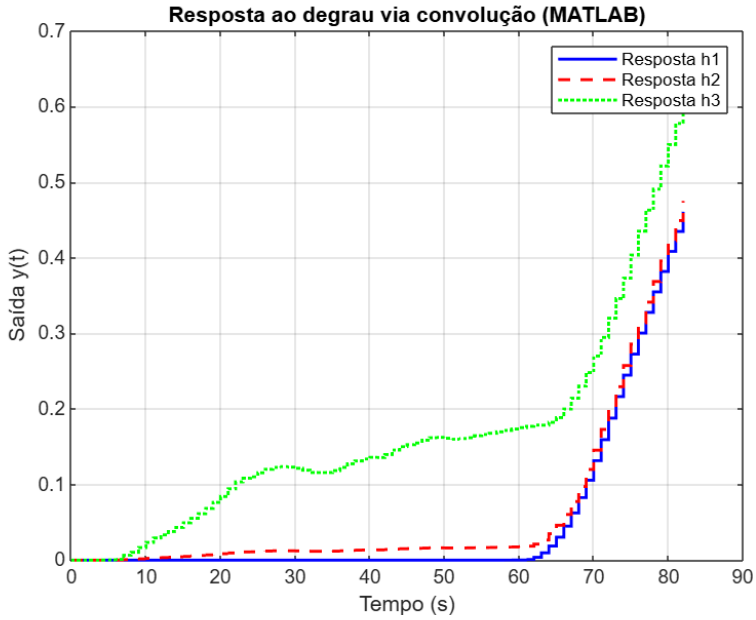


Figure 12: Curvas geradas após operação de convolução.

Listing 8: Código MATLAB.

```
>>
T = 1;
t_sim = 0:T:20;
u = 0.1 * ones(size(t_sim));
>>
y_conv_h1 = conv(u, out.h1.Data, 'full');
y_conv_h1 = y_conv_h1(1:length(u));
y_conv_h2 = conv(u, out.h2, 'full');
y_conv_h2 = y_conv_h2(1:length(u));
y_conv_h3 = conv(u, out.h3, 'full');
y_conv_h3 = y_conv_h3(1:length(u));
```

9 i.

Pelo método de correlação foram gerados três estimativas, que podem ser vistas na figura 13. Neste caso, podemos observar que as estimativas realizadas mantém as tendências das curvas do exercício anterior mas não mantém a constância em 0 do período de 0 a 60.

10 j.

Ao aplicar a operação convolução com pulso degrau com as estimativas pelo método de correlação, obtemos as três curvas da figura 14. Nota-se que por mais que a estimativa gerou curvas que não fossem tão parecidas quanto às originais, a curva após a convolução é idêntica a convolução realizada na alternativa h.

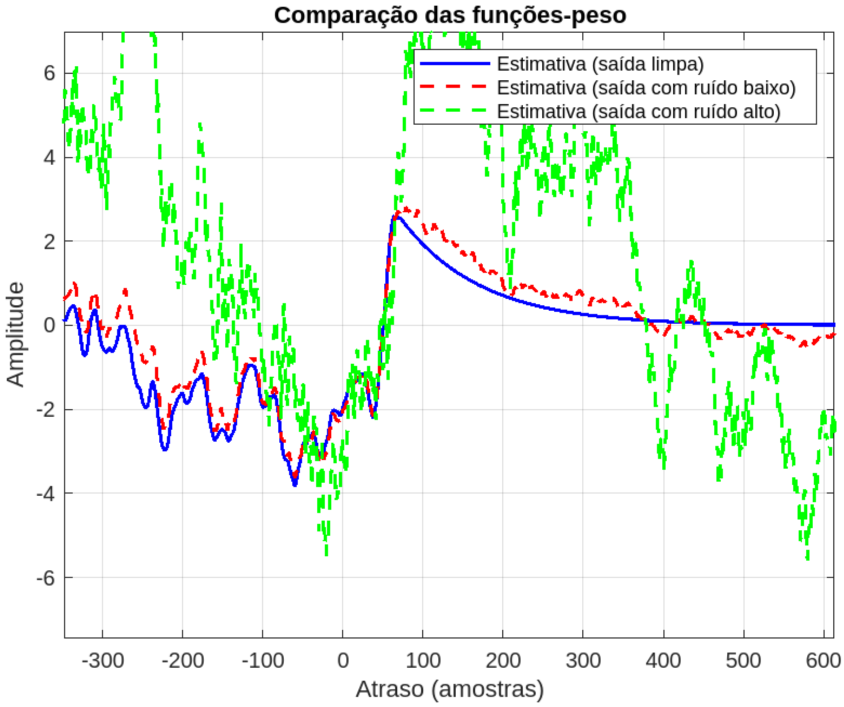


Figure 13: Estimativas

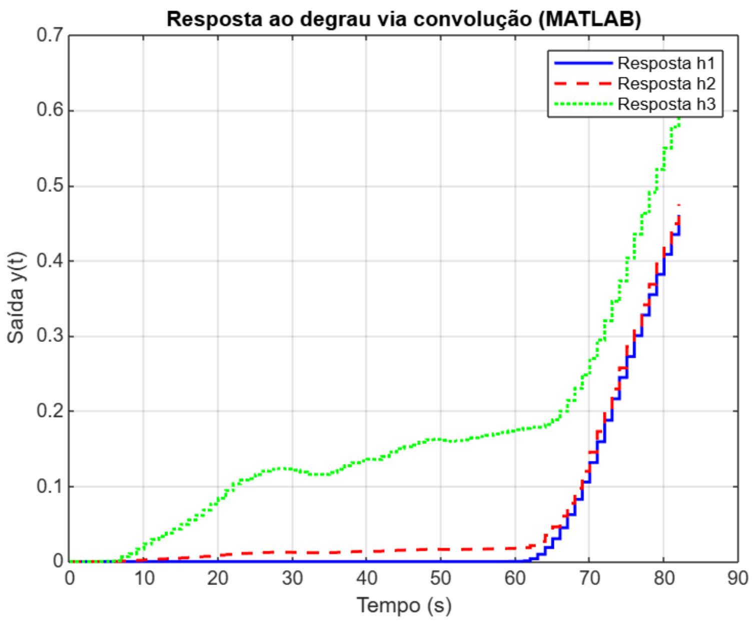


Figure 14: Convolução com estimativa de correlação