Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus João Pessoa
Tec. em Automação Industrial – TEC.0252 Controle Digital
Profº: Ademar Gonçalves da Costa Júnior, Dr.

PROJETO 3 - CONTROLE DIGITAL

Amadeus de Arandas - 20181450020
Carlos Daniel da Silva - 20201450011
Marcos Augusto Belinski - 20191450019
Samara Teixeira do Nascimento - 20192450005

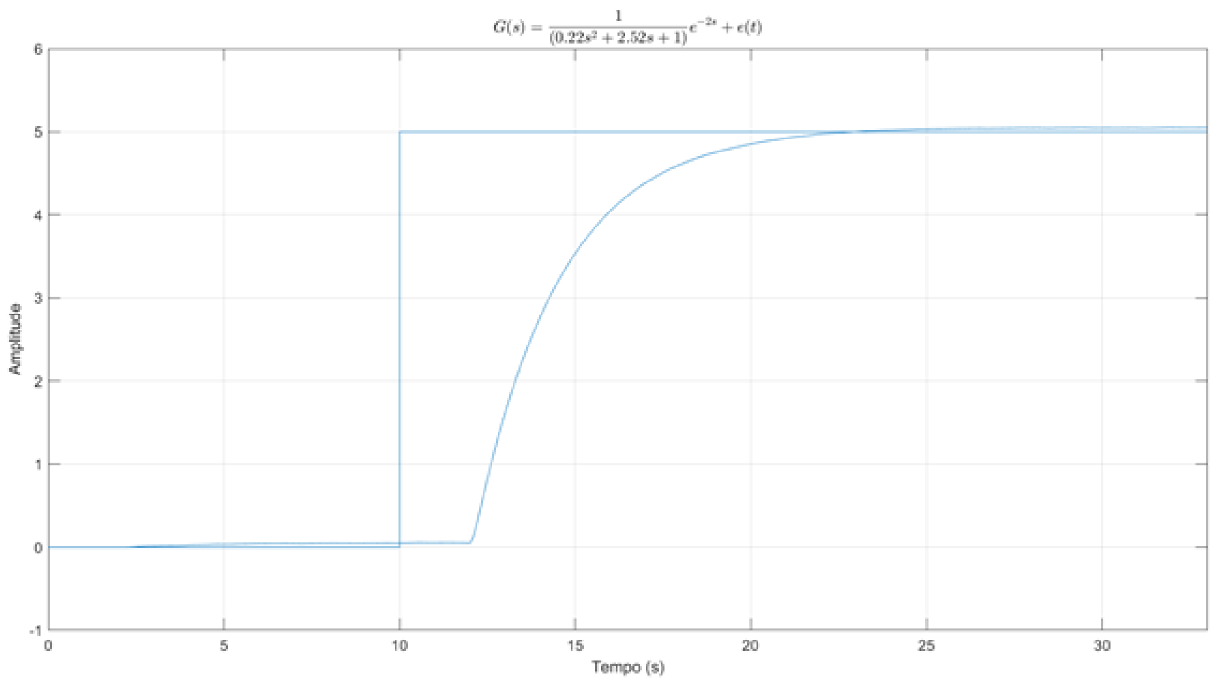
João Pessoa - PB

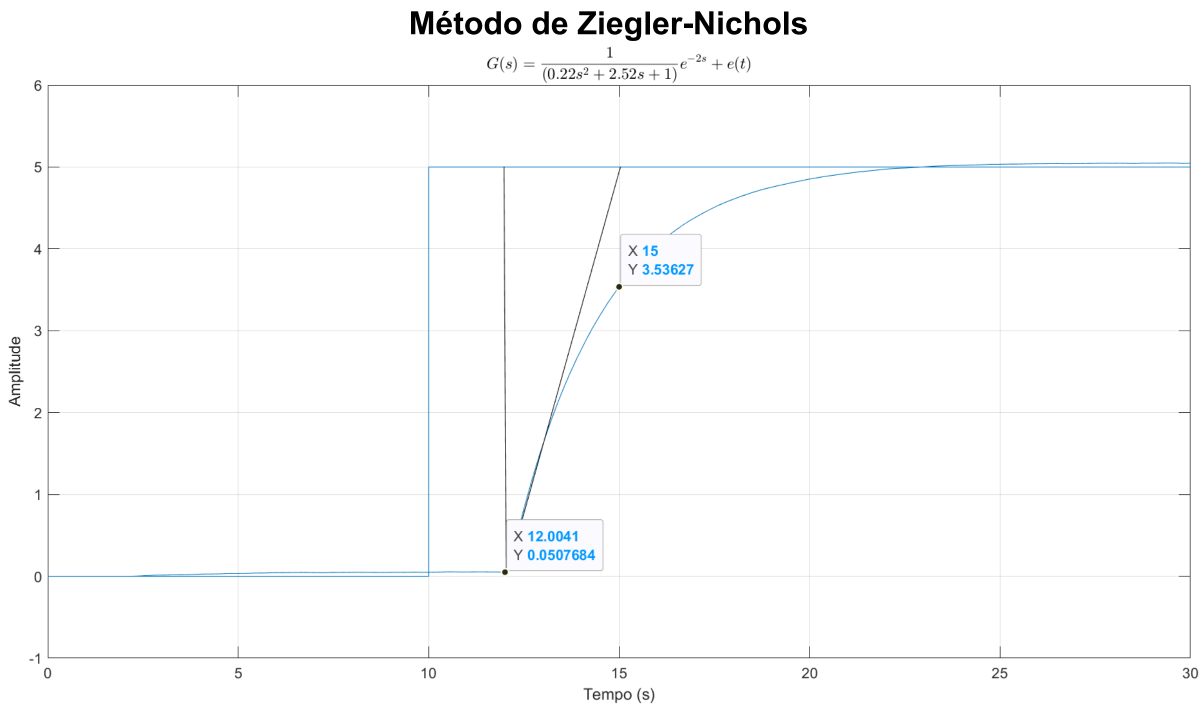
1. Com o uso de técnicas de identificação de modelos matemáticos, obtenham modelos matemáticos de primeira ordem e de segunda ordem que possam representar o comportamento matemático do circuito elétrico em questão (deverão usar os métodos apresentados, justificando seu uso ou não). Escolham uma faixa de operação entre 0 e 5 V e validem o modelo matemático utilizando índices estatísticos. Também deve ser justificado qual o tempo de amostragem a ser utilizado no projeto.

Faixa de tensão escolhida de 5 V

- Circuito simulado

$$G(s)=\frac{1}{0,22\,s^2+2,52\,s+1}e^{(-2s)}+e(t)$$





TD = 2
Tp = 3

segundos → Tempo de atraso obtido a partir do gráfico

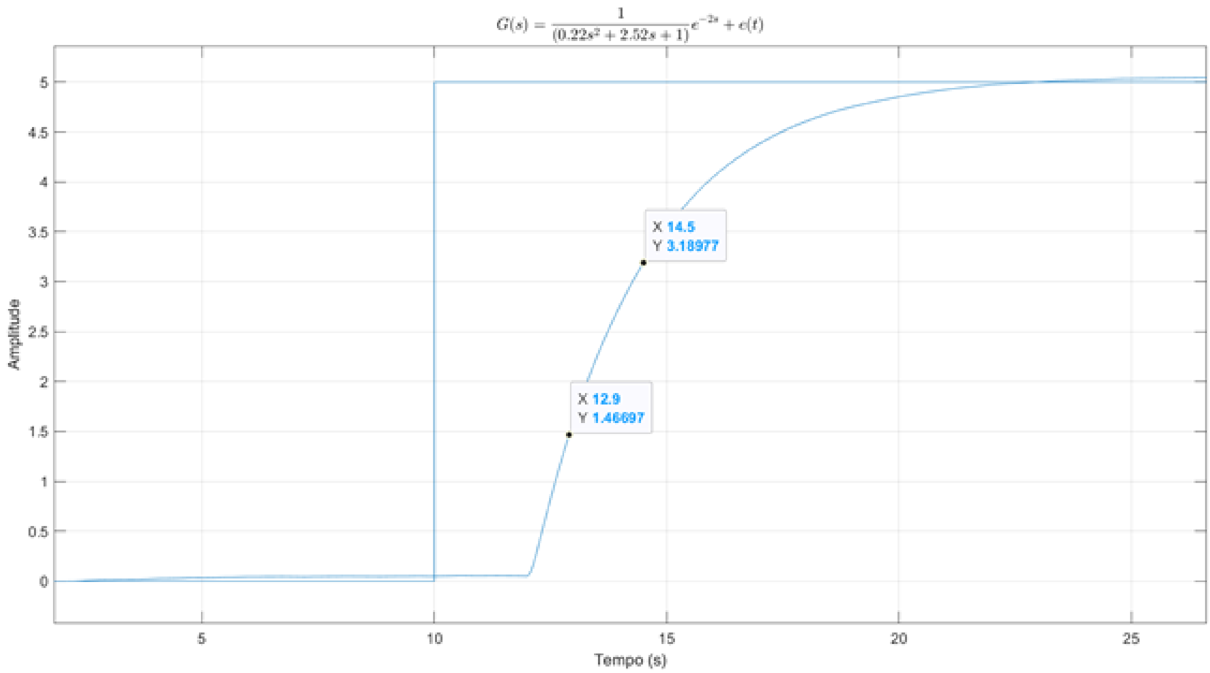
Td= 2 segundos → Tempo de atraso obtido a partir do gráfico

T= 3 segundos → Constante de tempo obtido através da reta tangente no gráfico

Substituindo os valores obtidos pelo gráfico na fórmula do método de Ziegler-Nichols, temos:

$$G(s)=\frac{Y(s)}{U(s)}=\frac{1}{3s+1}e^{(-2s)}=\frac{0{,}33}{s+0{,}33}e^{(-2s)}$$

Método de Smith de 1ª Ordem



$$G(s)=\frac{Y(s)}{U(s)}=\frac{K}{1+sT}e^{-sTd}$$

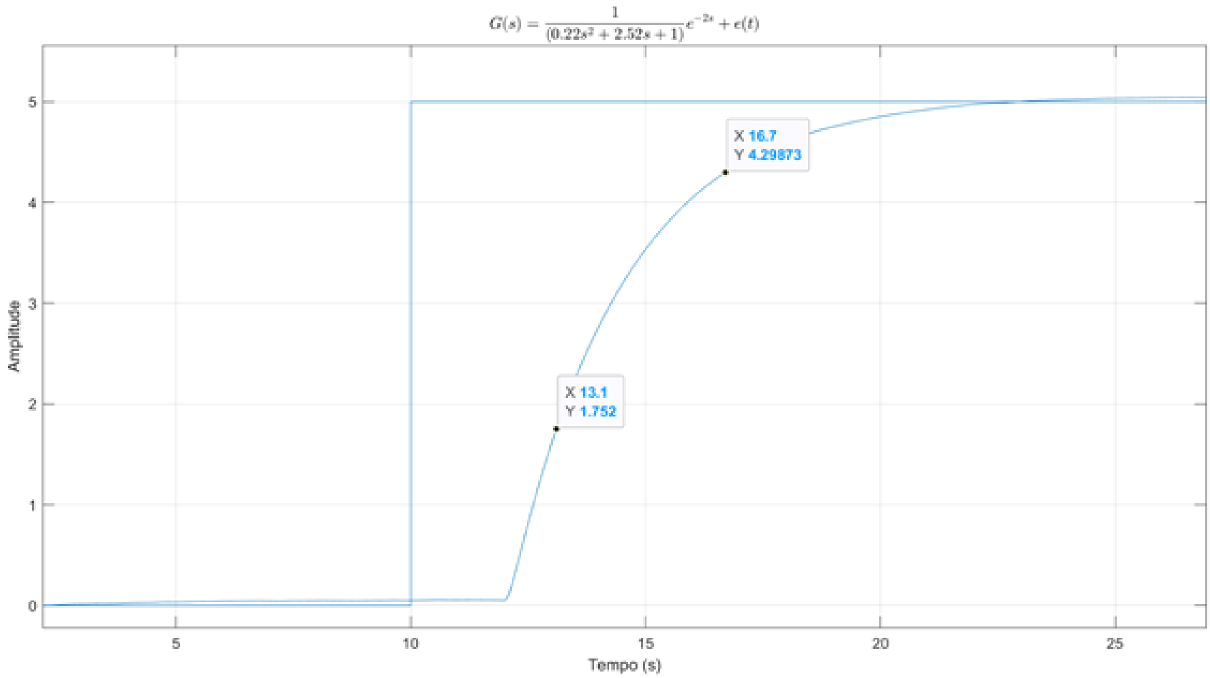
T = 1,5(14,5 – 12,9) = 1,5(1,6) = 2,4 segundos

T_D = (14,5 – 10) – 2,4 = 2,1 segundos

- Substituindo os valores obtidos pelo gráfico na fórmula do método de Smith de 1ª Ordem, temos:

$$G(s)=\frac{Y(s)}{U(s)}=\frac{1}{2{,}4s+1}e^{(-2{,}1s)}=\frac{0{,}42}{s+0{,}42}e^{(-2{,}1s)}$$

Método de Sundaresan-Krishnaswamy



$$G(s)=\frac{Y(s)}{U(s)}=\frac{K}{1+sT}e^{-sTd}$$

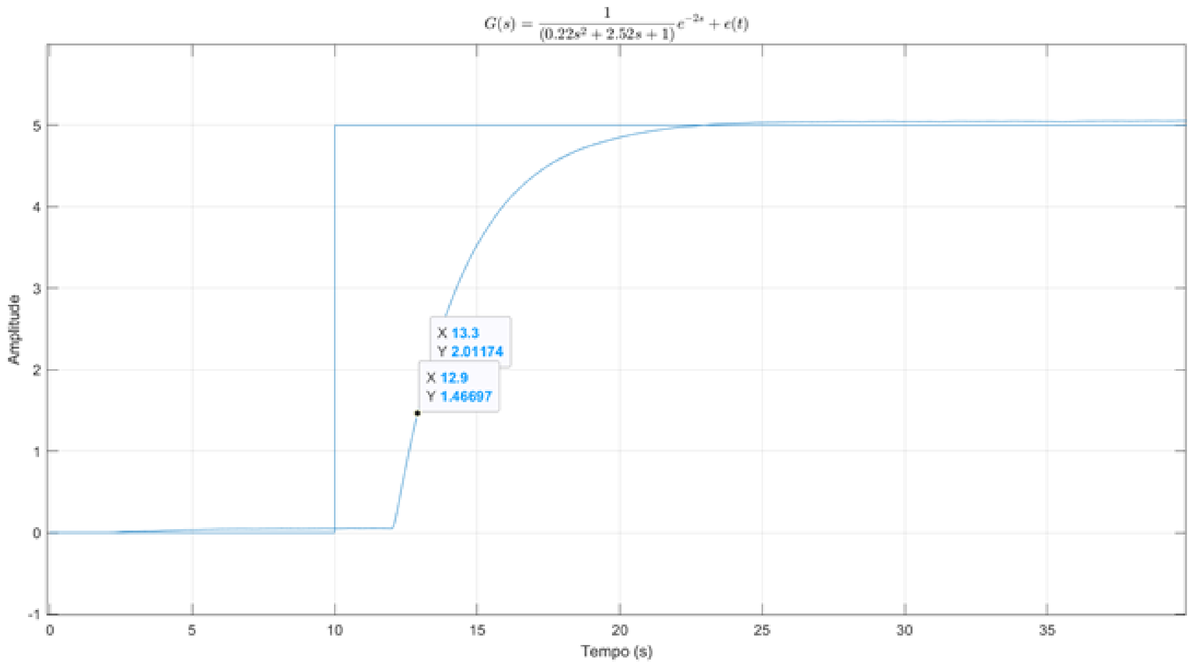
T = 0,67(16,7 – 13,1) ≅ 2,4 segundos

T_D = (1,3*3,1) – (0,29*6,7) ≅ 2,1 segundos

Substituindo os valores obtidos pelo gráfico na fórmula do método de Sundaresan-Krishnaswamy, temos:

$$G(s)=\frac{Y(s)}{U(s)}=\frac{1}{2{,}4s+1}e^{(-2{,}1s)}=\frac{0{,}42}{s+0{,}42}e^{(-2{,}1s)}$$

Método de Broida



$$G(s)=\frac{Y(s)}{U(s)}=\frac{K}{1+sT}e^{-sTd}$$

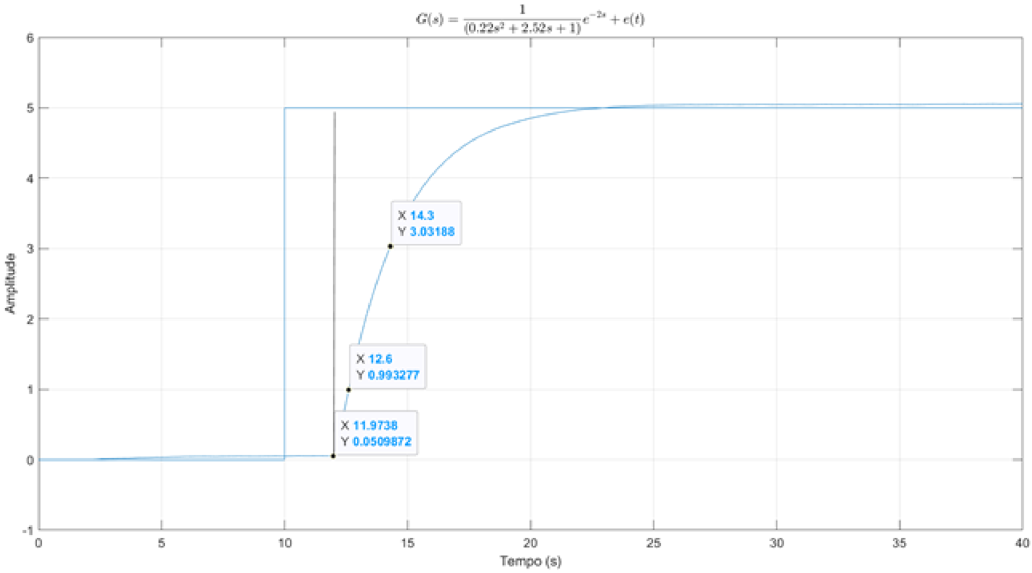
T = 5,5(13,3 – 12,9) = 2,2 segundos

T_D = (2,8*2,9) – (1,8*3,3) ≅ 2,2 segundos

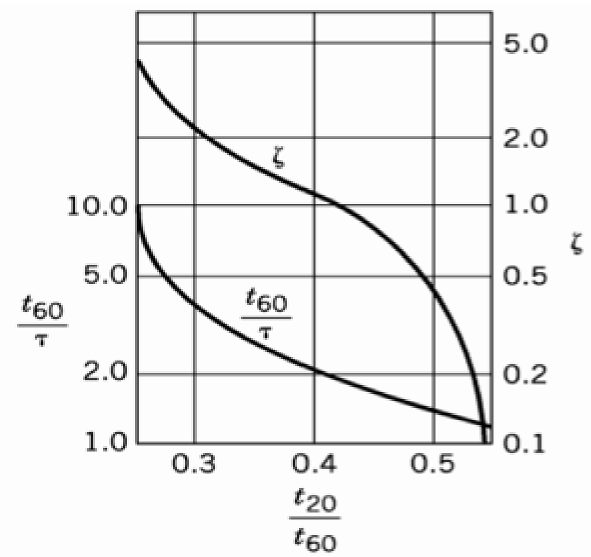
- Substituindo os valores obtidos pelo gráfico na fórmula do método de Broida, temos:

$$G(s)=\frac{Y(s)}{U(s)}=\frac{1}{2{,}2s+1}e^{(-2{,}2s)}=\frac{0{,}45}{s+0{,}45}e^{(-2{,}2s)}$$

Método de Smith de 2ª Ordem

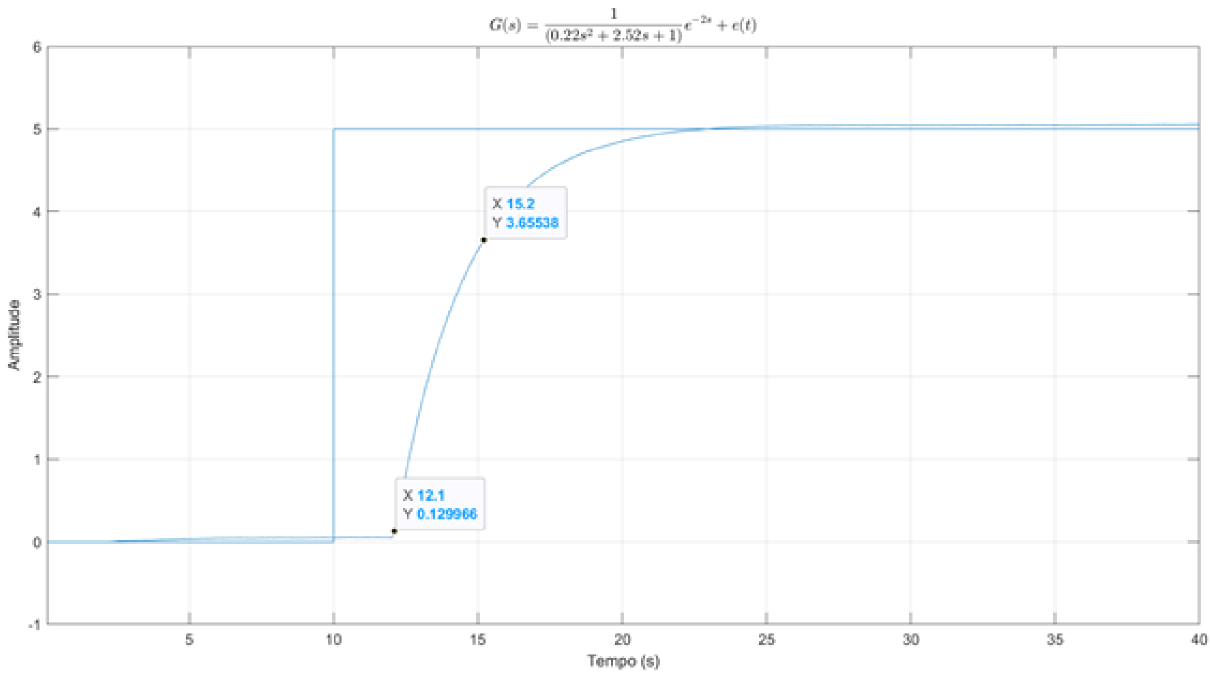


$t_{20\%}$ = 0,6 segundos ; $t_{60\%}$ = 2,3 segundos; $t_{20\%}/t_{60\%}$ = 0,26; ζ = 2,4;
$t_{60\%}/\tau$ = 4,2; τ = 1,83; TD = 2.1 segundos ;



$$G(s)=\frac{Y(s)}{U(s)}=\frac{K}{\tau s^2+2\zeta\tau s+1}e^{-(sTd)}=\frac{1}{1{,}83s^2+8{,}42s+1}e^{-(2{,}1s)}$$

Método de Harriot



t_0 = 12 – 12=0

$t_{73\%}$ = 15,2 – 12=3,2

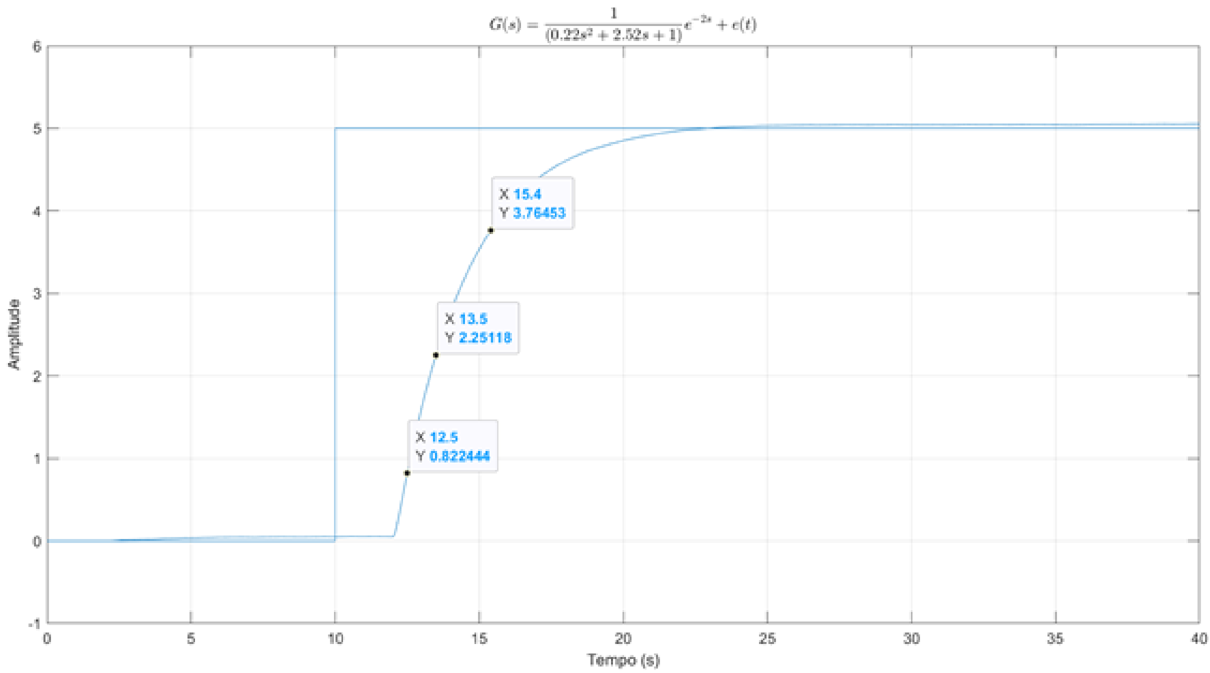
$$t1=t0+\frac{t73-t0}{2{,}6}=0+\frac{3{,}2}{2{,}6}=0{,}12\,segundos$$

- Utilizando o gráfico temos que y1 = 0,13, logo:

$$\frac{y1}{y\infty}=\frac{0{,}13}{5}=0{,}026$$

- Não foi possível estimar os valores de τ_{H1} e τ_{H2} utilizando a curva de Harriot, pois o valor de y_1/y_∞ foi menor que 0,2 que é o mínimo presente na curva de Harriot.

Método de Mollenkamp



t_1= 0,5 segundos

t_2 = 1,5 segundos

t_3 = 3,4 segundos

$$x = \frac{t2 - t1}{t3 - t1} = \frac{1{,}5 - 0{,}5}{3{,}4 - 0{,}5} = 0{,}34$$

$$\xi=\frac{0{,}0805-5{,}547(0{,}475-x)^2}{x-0{,}356}=\frac{0{,}0805-5{,}547(0{,}018)}{-0{,}016}=\frac{-0{,}019}{-0{,}016}=1{,}19$$

Como ξ≥1, temos:

$f_2(\xi)$ = 2,6ξ-0,6 = 2,49

$$\omega n=\frac{f\,2(\xi)}{t\,3-t\,1}=\frac{2{,}49}{2{,}9}=0{,}86\,rad/s$$

$$f\,3(\xi)=0{,}922(1{,}66)^{\xi}=0{,}922(1{,}66)^{1{,}19}=1{,}69$$

$$Td=t\,2-\frac{f\,3(\xi)}{\omega n}=1{,}5-\frac{1{,}69}{0{,}86}=-0{,}47$$

Como ξ ≥ 1, temos:

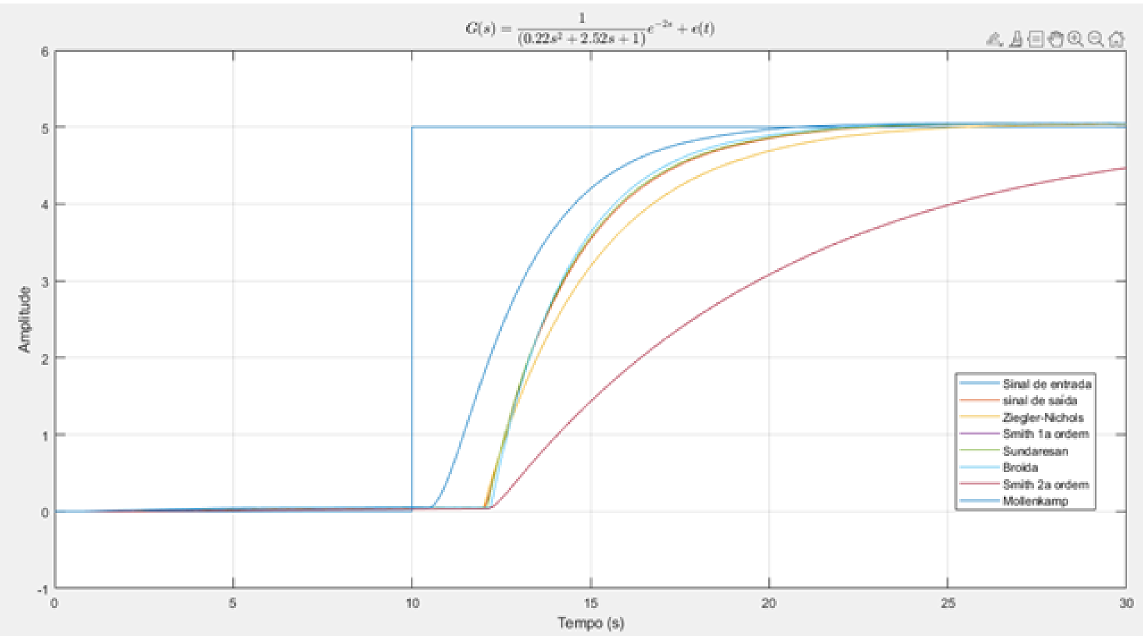
$$\tau 1{,}2=\frac{\xi\pm\sqrt{\xi^2-1}}{\omega n}$$

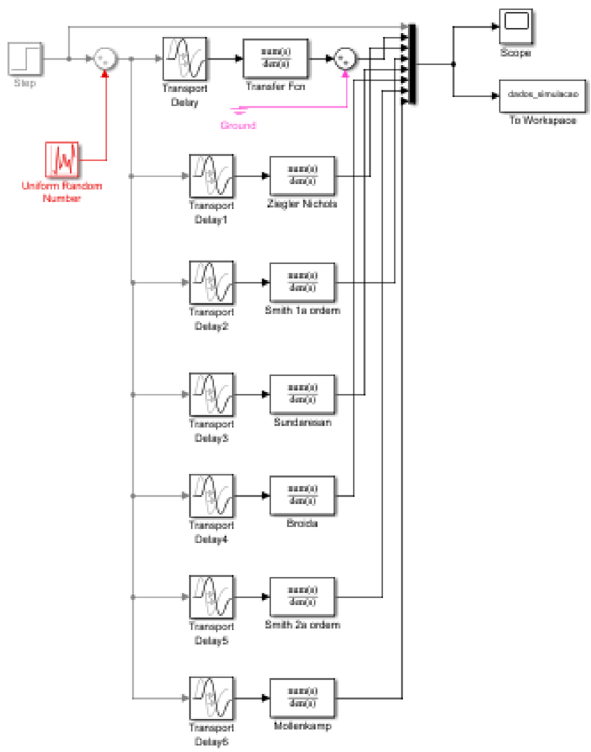
$$\tau 1=\frac{1{,}19+\sqrt{1{,}19^2-1}}{0{,}86}=2{,}1$$

Resultados com os modelos

Rotina no Matlab para plotagem dos modelos matemáticos:

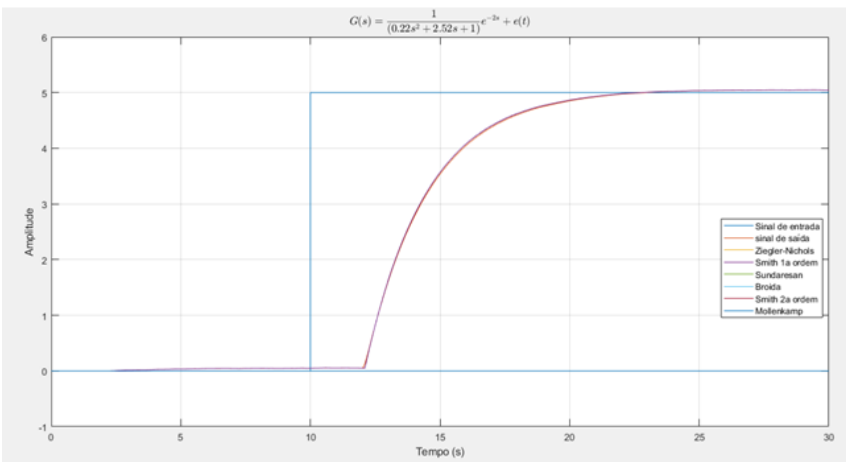
```
figure()

plot(tout,dados_simulacao)

legend('Sinal de entrada','sinal de saída','Ziegler-Nichols','Smith 1a ordem','Sundaresan','Broida','Smith 2a ordem','Mollenkamp')

axis([0 30 -1 6])

grid on

xlabel('Tempo (s)')

ylabel('Amplitude')

title('$$G(s)=\frac{1}{(0.22s^2+2.52s+1)}e^{-2s}+e(t)$$','interpreter','latex')
```

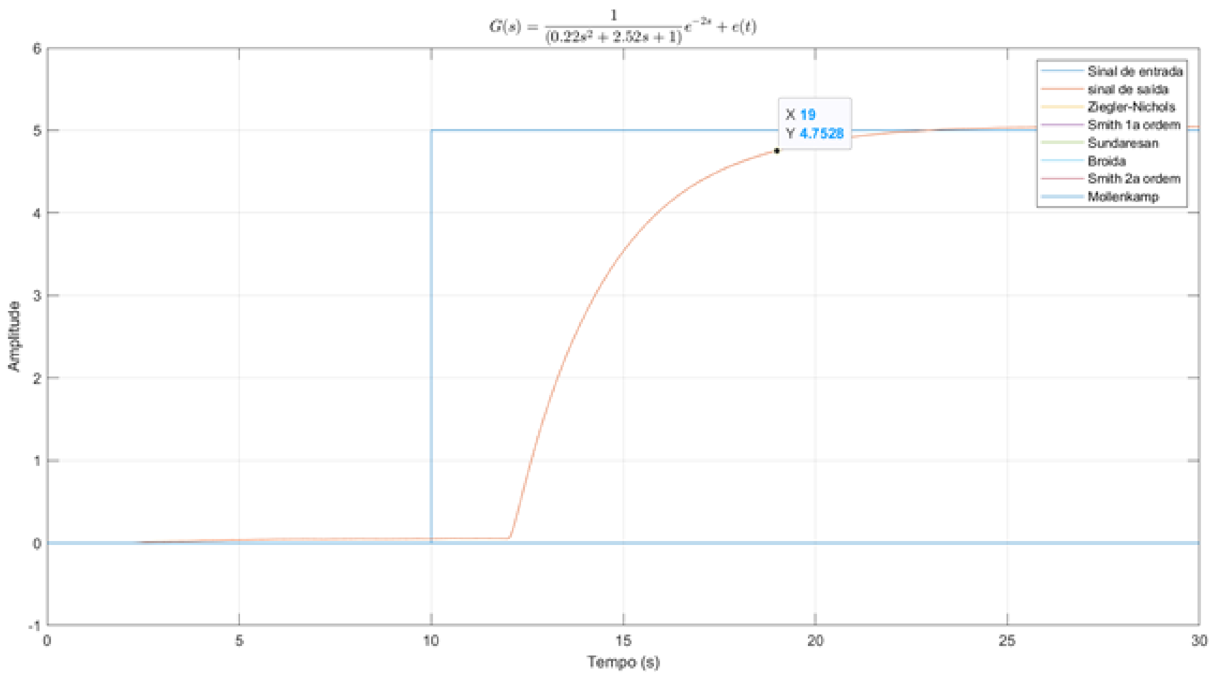




- Com a simulação dos métodos, pudemos constatar que o melhor método para representar o comportamento matemático do circuito elétrico é o método de smith 1ª ordem, pois foi o que mais se aproximou graficamente da função G(s) calculada.

Tempo de Amostragem





$t_{95\%}$ = 7 segundos

Determinando o tempo de amostragem pelo método de Isermann & Munchhof:

$$\frac{t\,95\%}{15} \leq ta \leq \frac{t\,95\%}{5}$$

0,47≤ta≤1,4 segundos

Sendo assim, por consenso foi decidido utilizar o tempo de amostragem de 0,5 segundos.

3. Projete um controlador PI e PID (na ação derivativa, incluam o filtro), utilizando a arquitetura ideal e cada método apresentado (ver item 8.1b). Utilizem o modelo estimado do item 1 e o modelo fenomenológico (ver projeto 1) para a sintonia dos parâmetros do controlador PID. Avaliem os controladores com índices qualitativos (IAE e ITAE).

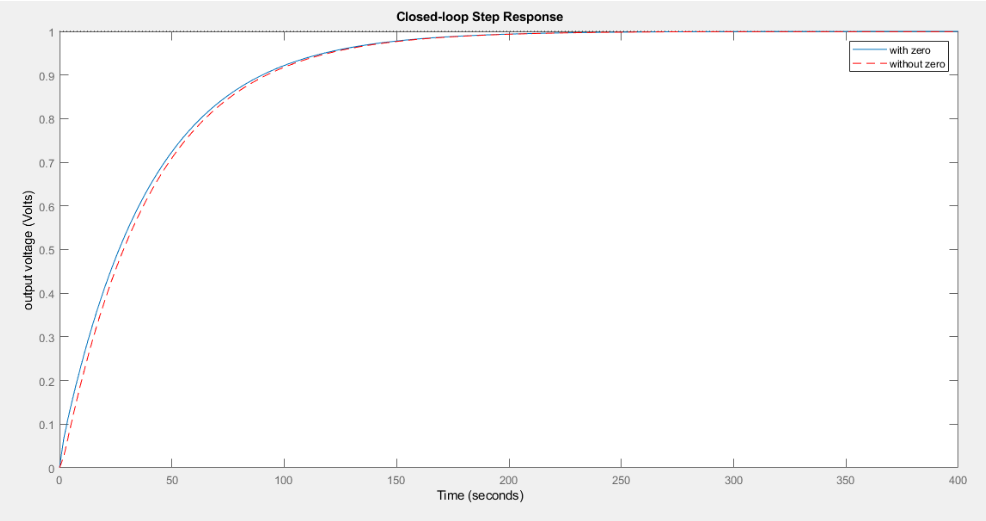
Controlador PI

Rotina de Programação

```
%{
simulação com circuito RC.
a =  Circuit_RC.signals.values(:,1);
b = Circuit_RC.signals.values(:,2);
tempo = Circuit_RC.time;
figure(), plot(tempo,a),hold on, plot(tempo,b)
xlabel('time (seconds)')
ylabel('voltage (Volts)')
title('RC Circuit Step Response')
axis([0 5 0 2.5])
%}
%{
teoria da resposta da frequencia
s = tf('s');
```

```
R = 320000;                          % resistencia  RC circuit
C = 101*10^-6;                       % capacitancia no RC circuit
G = 1/(C*R*s+1);                     % RC circuit função transferencia
bode(G)
title('RC Circuit Frequency Response (R = 320 kOhm, C = 101 uF)')
%}
%{
aquisição de dados no simulink
T=4;      % Período da entrada da onda quadrada
Ts=0,05;  % de tempo de amostragem empregado no modelo Simulink
N=T/Ts;   % Número de amostras por período
%}
```

Gráfico Gerado



Obs: Para algumas partes requer a plotagem de novos gráficos (Sis: PI)

Controlador PID

Rotina de Programação

%{

Vamos primeiro ver a resposta ao degrau em malha aberta.

s = tf('s');

P = 1/(0.22*s^2 + 2.52*s + 1);

step(P)

%}

%{

A partir da tabela mostrada acima, vemos que o controlador proporcional (K_p) reduz o tempo de subida, aumenta o overshoot e reduz o erro em regime permanente.

A função de transferência de malha fechada do nosso sistema de feedback unitário com um controlador proporcional é a seguinte, onde X(s)é nossa saída (igual Y(s)a) e nossa referência R(s)é a entrada:

Kp = 300;

C = pid(Kp)

T = feedback(C*P,1)

t = 0:0,01:2;

passo(T,t)

%}

%{

Agora, vamos dar uma olhada no controle PD. A partir da tabela mostrada acima, vemos que a adição do controle derivativo (K_d) tende a reduzir tanto o overshoot quanto o tempo de acomodação. A função de transferência em malha fechada do sistema dado com um controlador PD é:

Kp = 300;

Kd = 10;

C = pid(Kp,0,Kd)

T = feedback(C*P,1)

t = 0:0,01:2;

passo(T,t)

%}

%{

Antes de prosseguir com o controle PID

vamos investigar o controle PI.

A partir da tabela, vemos que a adição de controle integral (Ki) tende a diminuir o tempo de subida

Aumentar tanto o overshoot quanto o tempo de estabilização e reduzir o erro em regime permanente.

Para o sistema dado, a função de transferência em malha fechada com um controlador PI

Kp = 30;

Ki = 70;

C = pid(Kp,Ki)

T = feedback(C*P,1)

t = 0:0,01:2;

passo(T,t)

%}

%{

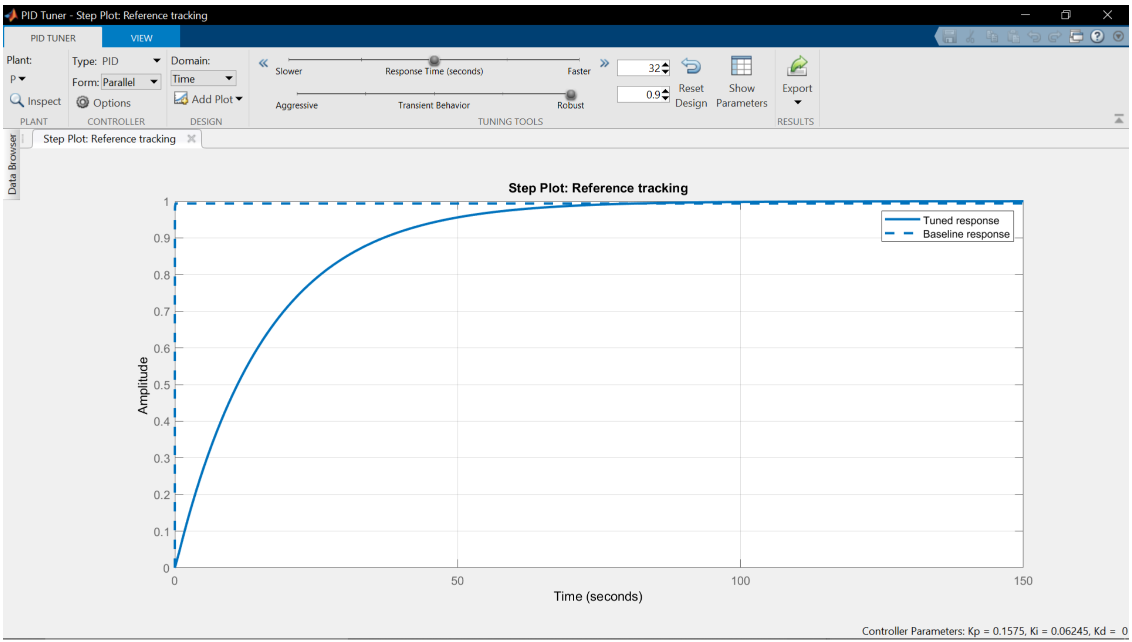
Agora, vamos examinar o controle PID. A função de transferência em malha fechada do sistema dado com um controlador PID

Após várias iterações de ajuste, os ganhos Kp= 350, Ki= 300 e Kd= 50 forneceram a resposta desejada.

Kp = 350;

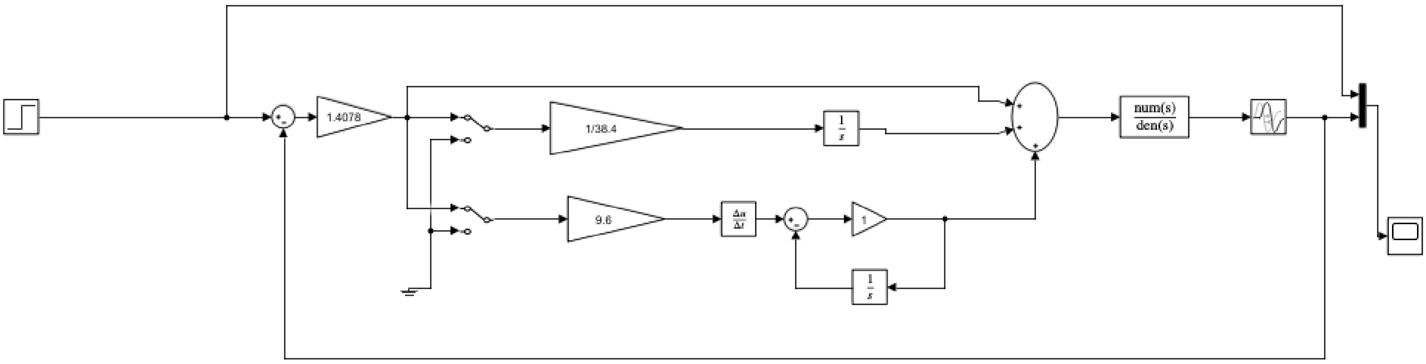
Ki = 300;

Gráfico Gerado



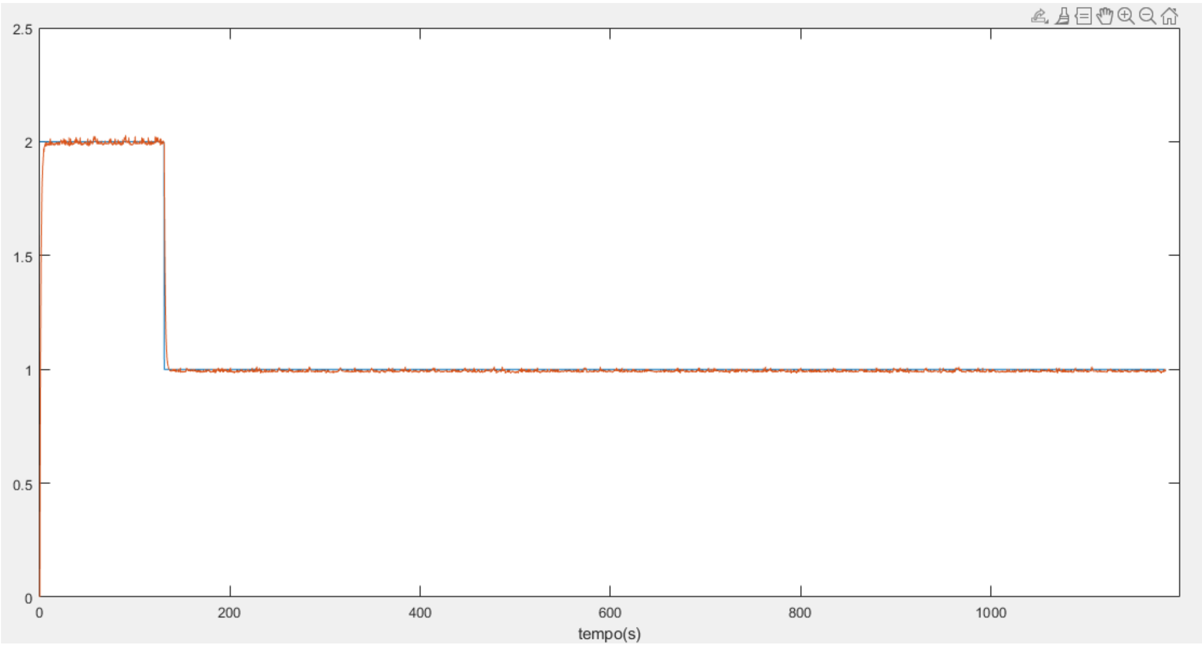
Obs: Para algumas partes requer a plotagem de novos gráficos (Sis: PID)

Elaboração do Circuito PID via Simulink



Simulação com Circuito RC

```
a=Circuit_RC.signals.values(:,1);

b=Circuit_RC.signals.values(:,2);

tempo = Circuit_RC.time;

figure(), plot(tempo,a),hold on, plot(tempo,b)

xlabel('tempo(s)')
```



Circuito RC via Simulink