Aula 7 - Laboratório de Controle - 2023/1

Modelagem e controle de um sistema de ordem 1 (circuito RC)

Nomes: João Gabriel, Lucas Tessari

Início

Procedimento:

1) Inicie o Matlab

2) Ligue o Arduíno numa portal USB (deve haver uma indicação disto)

3) Execute o script abaixo para identificar porta serial do Arduino e conectar com o Matlab.

Não dê o comando clear, pois apaga a variável obj gerada abaixo e impede a comunicação.

Caso a comunicação Matlab - Arduíno pare de funcionar, feche o Matlab e repita os passos acima.

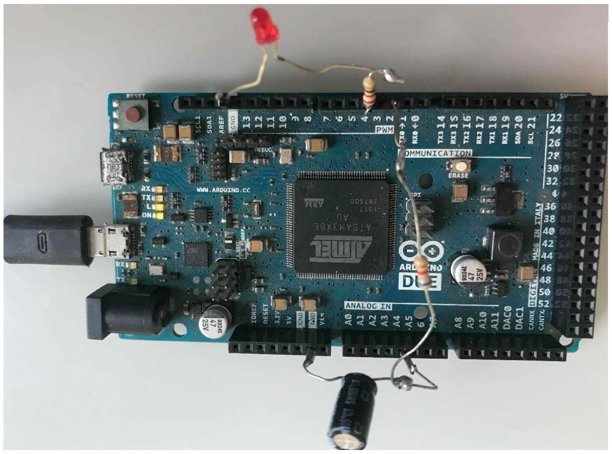
```
if ~exist('obj')
    z=seriallist;
    comPort=z{length(z)};
    obj=serial(comPort,'BaudRate',9600);
    obj.Terminator='CR';
    fopen(obj);
end
datetime
```

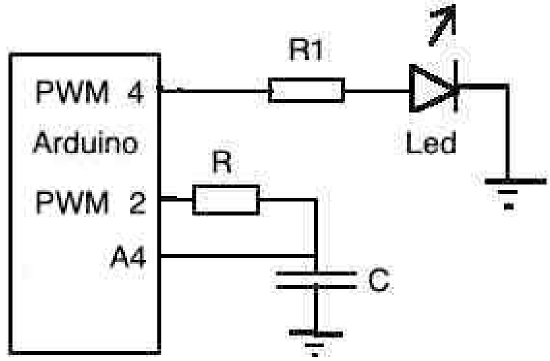
```
ans = datetime
   03-May-2016 08:48:14
```

Ligações do Arduíno usadas para esta aula

Usaramos o mesmo ambiente da aula 2. Um sinal PWM do Arduíno é aplicado no pino 2, gerando uma tensão média no capacitor C, que é lida pelo conversor A/D no pino A4.

O mesmo sinal PWM aplicado no capacitor é aplicado no led via resistor R1, para que seu brilho seja proporcional à tensão no capacitor. O sinal PWM aplicado tem 8bits e varia de 0 a 225, gerando uma tensão média no capacitor que varia de 0 a 3.3V. O conversor A/D converte a tensão no capacitor de 0 a 3.3V em um número inteiro de 0 a 1023. Logo, a entrada varia de [0, 255] e a saída de [0, 1023].

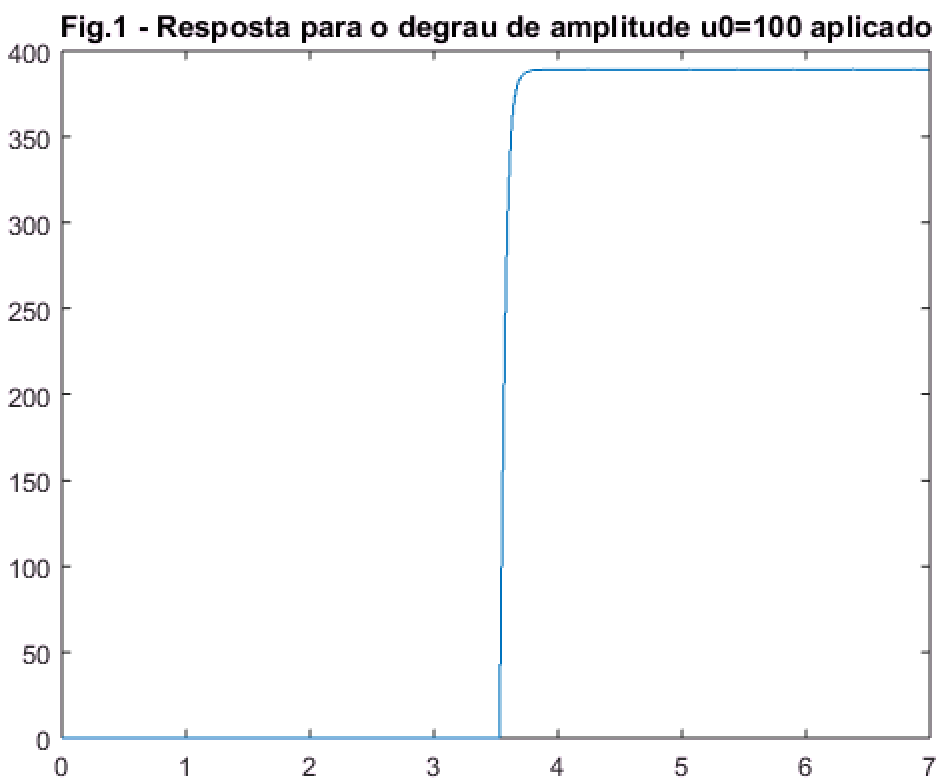




Atividade 1

Dar degrau de amplitude U0 e coletar a resposta do circuito RC usando o Arduino. O tempo de amostragem Ts é fixo em 20ms.

```
Tempo=7; % Escolher o tempo adequado para a visualizacao da resposta transitoria e de regime
U0=100;
[y,t]=arduino_coleta_rc(obj,0,Tempo); % zerar saida
[y,t]=arduino_coleta_rc(obj,[0 U0],Tempo);
figure
stairs(t,y);
ss=sprintf('Fig.1 - Resposta para o degrau de amplitude u0=%d aplicado',U0);
title(ss);
```




```
seta_saida_rc(obj,0)
```

Obtenha o modelo de ordem 1 $G(s) = \frac{K}{\tau s + 1}$ usando a função procest e verifique se o modelo representa bem os dados.

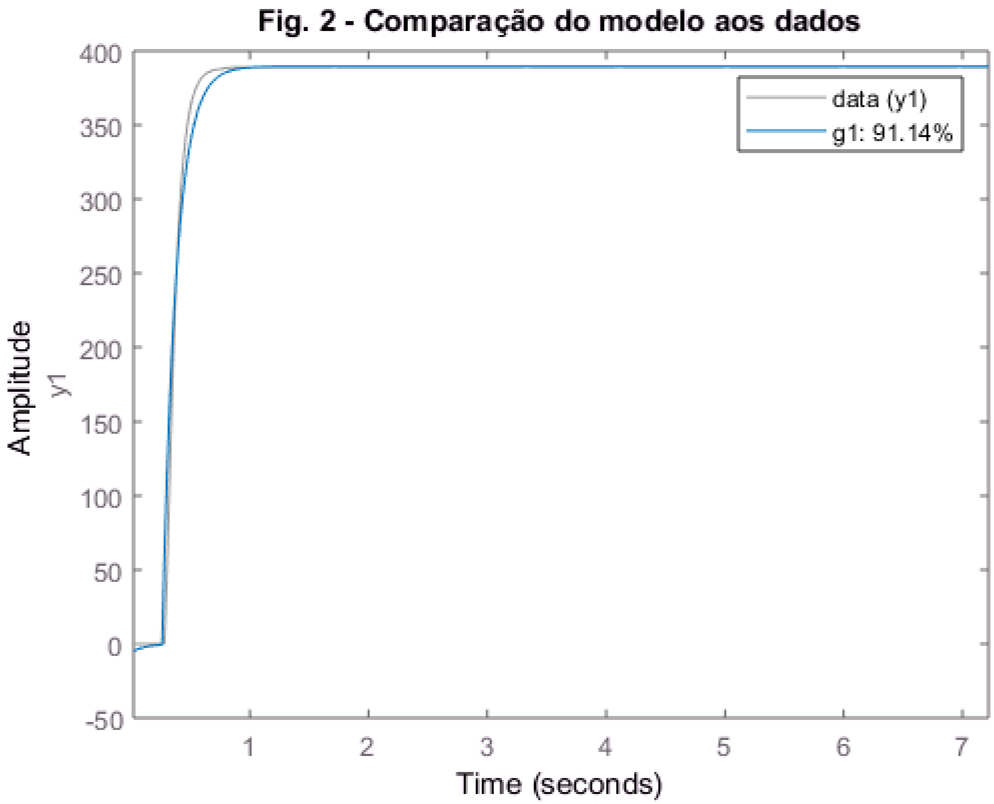
```
Ts=20; % Tempo de amostragem fixo em ms
N=floor(length(y)/2);
L=floor(N-10);
y0=mean(y(L-1:L));
yf=mean(y(end-5:end));
yn=y(L:end)-y0;
u=[zeros(size(y(L:N+1)));ones(N-1,1)*U0];
t=(0:length(yn)-1)*Ts/1000;
data=iddata(yn,u,Ts/1000); % Dados usados para obter o modelo
g1=procest(data,'P1'); % Função para estimar o modelo de ordem 1
G1=tf(g1.Kp,[g1.Tp1 1])
```

```
G1 =

      3.896
  ------------
  0.1174 s + 1

Continuous-time transfer function.
```

```
figure;
compare(data,g1);
title('Fig. 2 - Comparação do modelo aos dados');
```



1.1 Comente a semelhança da curva coletada e a obtida pelo modelo obtido, tanto no transitório quanto em regime, justificando diferenças.

Resposta: Ambas possuem o mesmo valor em regime permanente e um comportamento muito semelhante no regime transitório, salvo o fato do que a resposta modelada é mais lenta devido ao fato de que temos que levar em conta o atraso de processamento dos dados pelo hardware.

Atividade 2 - Projeto de um controlador PI via método lambda

Projetar um controlador PI via método lambda (da aula anterior) de modo a ter constante de tempo de malha fechada igual à de malha aberta.

Dada a FT de MA $G_p = \frac{K}{\tau s + 1}$ e o controlador $C(s) = K_p + \frac{K_p K_i}{s}$ com $K_p = \frac{\tau}{K\lambda}$ e $T_i = \frac{1}{K_i} = \tau$, resulta FT de MF

$$M(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} = \frac{1}{\lambda s + 1}.$$

Abaixo esse controlador é calculado e testado.

```
[y,t]=arduino_coleta_rc(obj,0,Tempo); % zerar saida
Ref=300;
tau=g1.Tp1;
K=g1.Kp;
```

```
lambda=tau; %  lambda = cte de tempo de MA
Kp=tau/(K*lambda);
Ki=1/tau;
[y,u,t] = arduino_controle_PI_rc(obj,Ref,Tempo, Kp, Ki);
plot(t,y,t,u);legend('y','u');
line([0 max(t)],[Ref Ref],'Color','red','LineWidth',2);
xlabel('Tempo(s)');
ss=sprintf('Fig. 3 - Resposta em malha fechada usando lambda = %3.2f',tau);
title(ss)
```


```
seta_saida_rc(obj,0);
```

2.1 O que é lambda? Compare a constante de tempo de malha fechada na Figura 3 com lambda, explicando.

Lambda e Tau possuem o mesmo valor igual a 0,1180. Lambda funciona como uma constante de tempo para malha fechada.

2.2 Descreva o comportamento do sinal de controle e sua proximidade aos limites de sua saturação (quais são eles?).

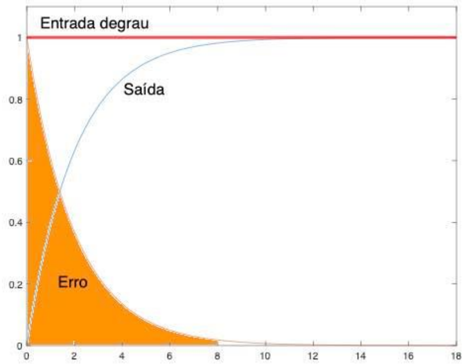
O sinal de controle se estabiliza em um limite (78) que é o valor necessário para que o produto entre o mesmo e a função de transferência de malha aberta

seja 300 (regime permanente). O sinal de controle se estabilizou longe de seus limites 0 e 255 visto que é um sinal pwm.

Atividade 3 - Efeito do parâmetro lambda no IAE

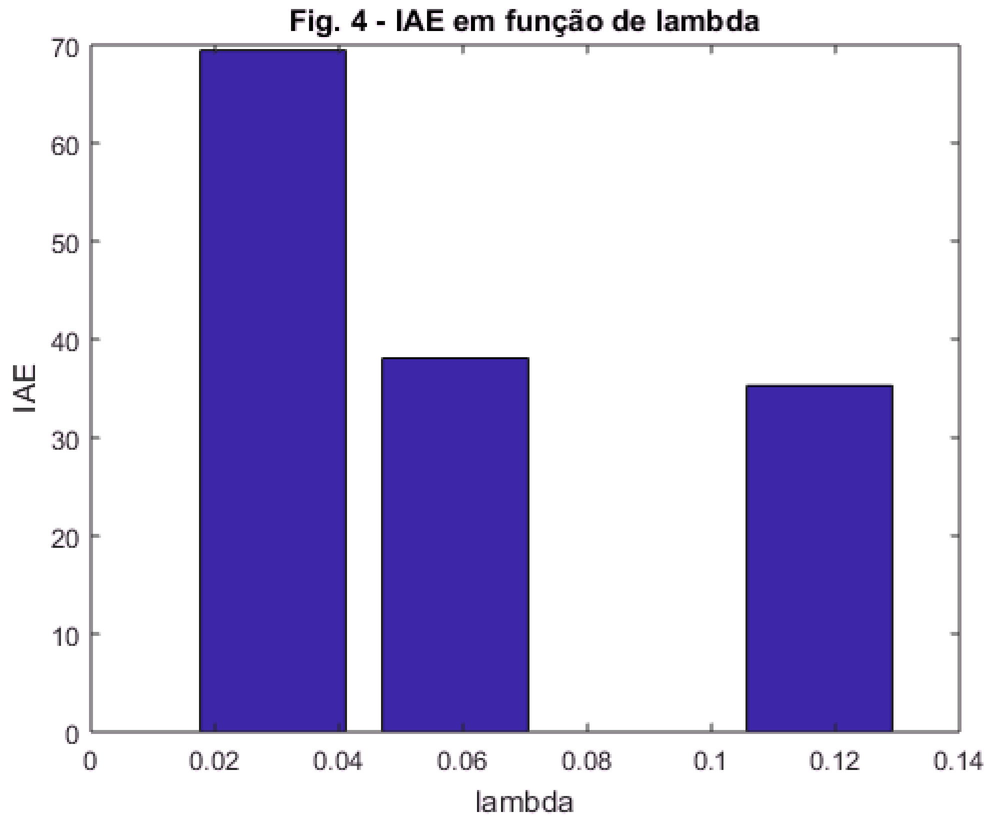
A Figura abaixo mostra o IAE: a área sob a curva do valor absoluto do erro.

Quanto mais lenta a resposta maior o IAE.



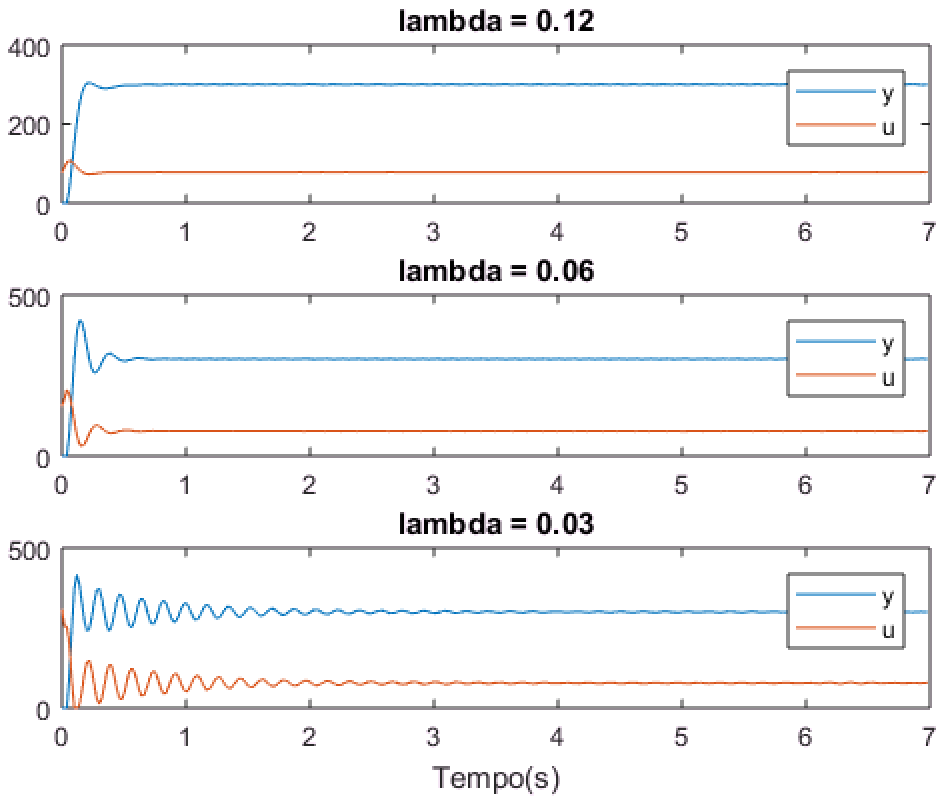
Abaixo são escolhidos 3 valores para lambda de modo a avaliar seu efeito sobre o IAE.

```
lambda=tau./[1 2 4];
for i=1:3
    Kp=tau/(K*lambda(i));
    Ki=1/tau;
    [y,u,t] = arduino_controle_PI_rc(obj,Ref,Tempo, Kp, Ki);
    erro=Ref-y;
    iae(i,1)=trapz(t,abs(erro));
    Y(i).y=y;
    Y(i).u=u;
    Y(i).t=t;
    seta_saida_rc(obj,0);
    pause(3);
end
figure;
bar(lambda,iae);xlabel('lambda');ylabel('IAE');
title('Fig. 4 - IAE em função de lambda')
```



A seguir são apresentadas as respostas ao degrau para essas três sintonias.

```
figure;
for i=1:3
    subplot(3,1,i);
    plot(Y(i).t,Y(i).y,Y(i).t,Y(i).u);legend('y','u');
    ss=sprintf('lambda = %0.2f',lambda(i));title(ss);
end
xlabel('Tempo(s)');
```



3.1 Qual foi o valor mínimo de IAE obtido e para que valor de lambda? Por que não ficou menor?

O valor mínimo de IAE foi de 34,19 com o valor de lambda igual a 0,118.

Vemos que o IAE fica maio isto que a resposta começa a oscilar aumentando o valor da integral do erro absoluto.

3.2 Compare o sinal de controle para lambda mínimo e máximo.

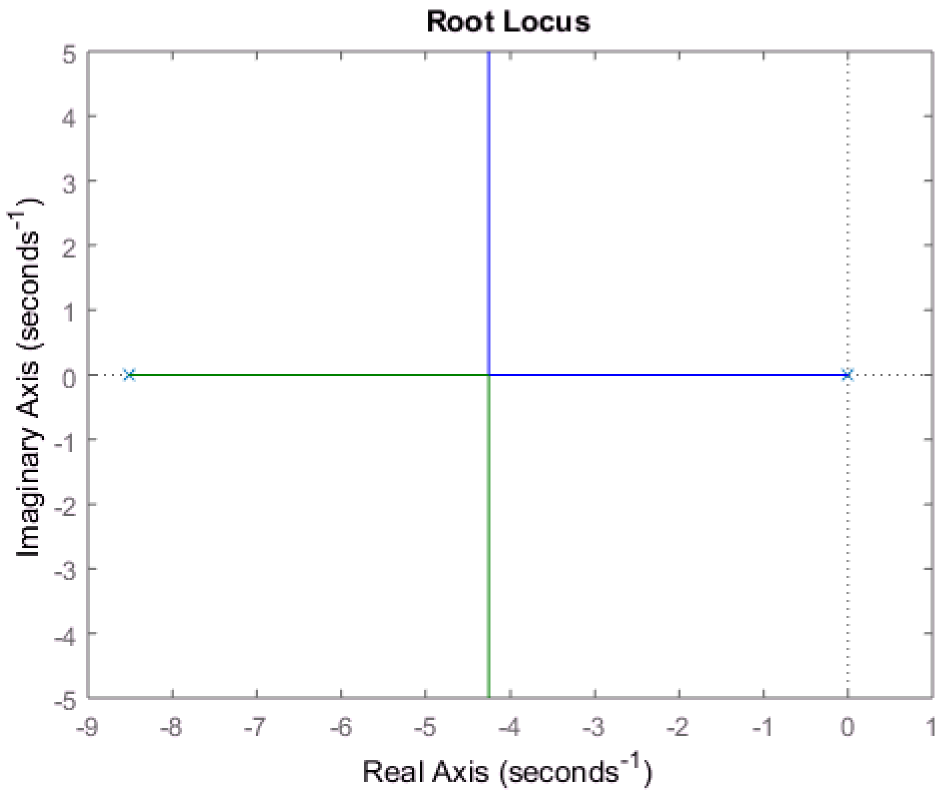
Para valores pequenos de lambda, nota-se que a resposta é muito oscilatória, aumentando o valor de lambda até ao valor da constante de tempo temos

em 0,12 o valor de lambda com o menor IAE. Vemos que isso acontee visto que o sinal de controle tenta estabilizar o sinal de resposta,logo se a mesma é oscilatória, o controlador tambem deve gerar um sinal oscilatório para estabilizar a resposta.

Atividade 4: Projeto do controlador PI via LR

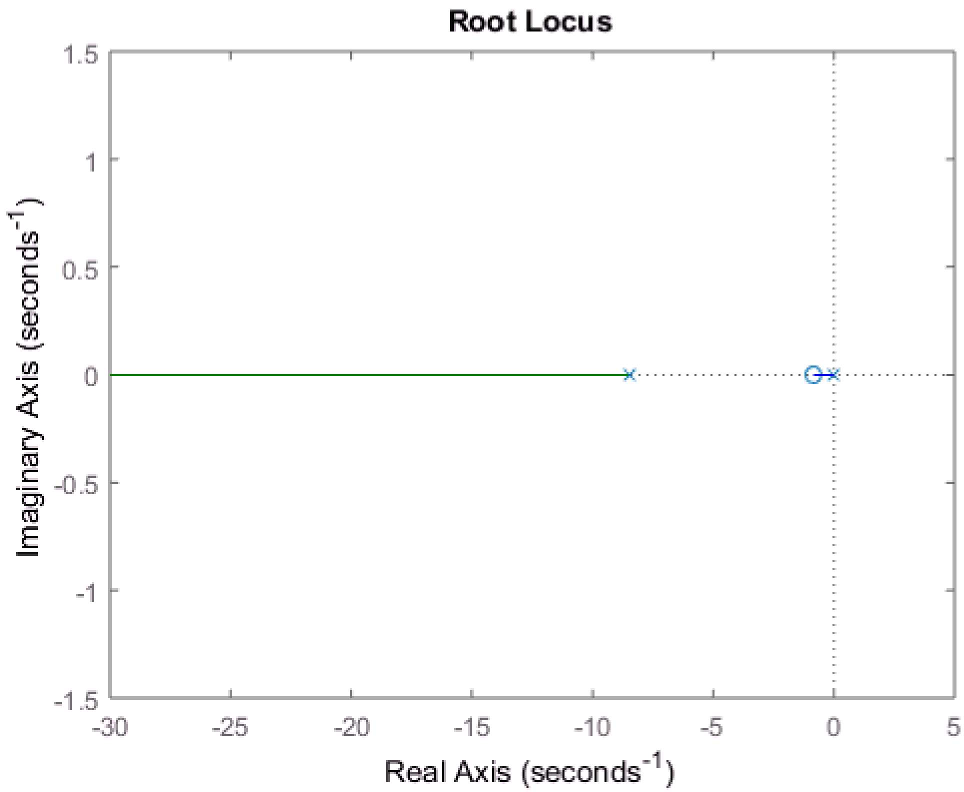
O controlador PI série tem a forma $C(s) = K_p + \frac{K_pK_i}{s} = K_p(\frac{s + K_i}{s})$, ou seja, um zero em $s = -K_i$ com o ganho K_p. Ao adicionar o polo na origem do controlador PI na FT G1(s), o LR fica da forma abaixo,

```
figure;
rlocus(G1*tf(1,[1 0]));
```

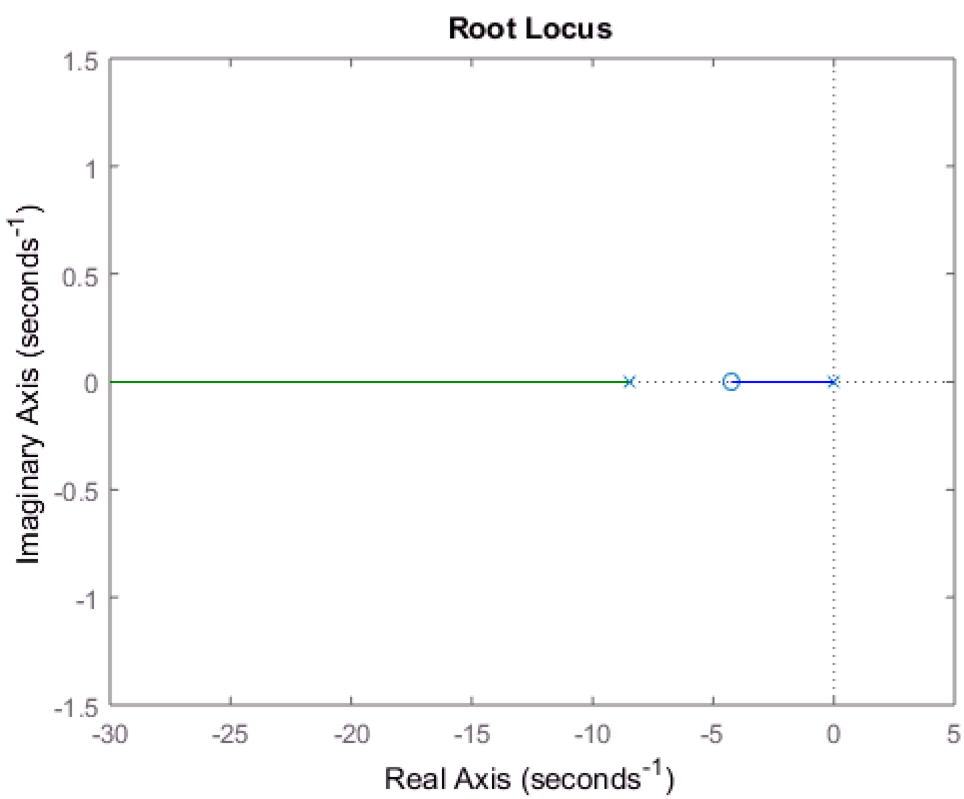


Para o projeto, o zero do PI deve ser adicionado sobre o eixo real, sendo mostradas algumas alternativas abaixo.

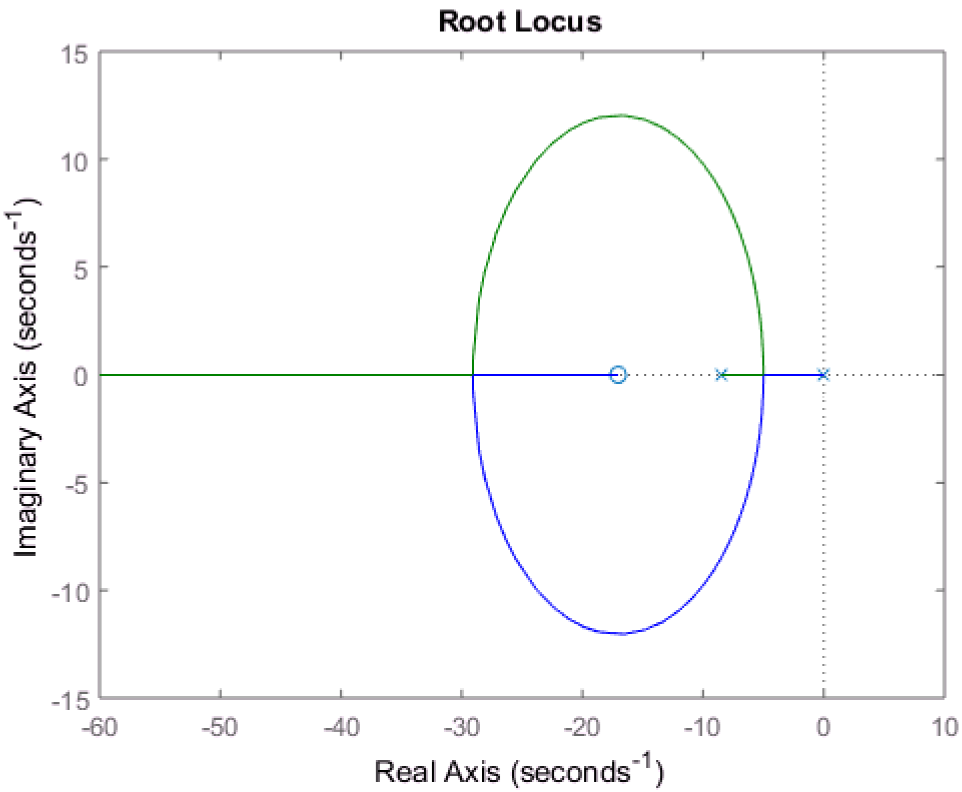
```
pl=pole(G1);
figure;rlocus(G1*tf([1 -pl/10],[1 0]));
```




```
figure;rlocus(G1*tf([1 -pl/2],[1 0]));
```




```
figure;rlocus(G1*tf([1 -2*pl],[1 0]));
```



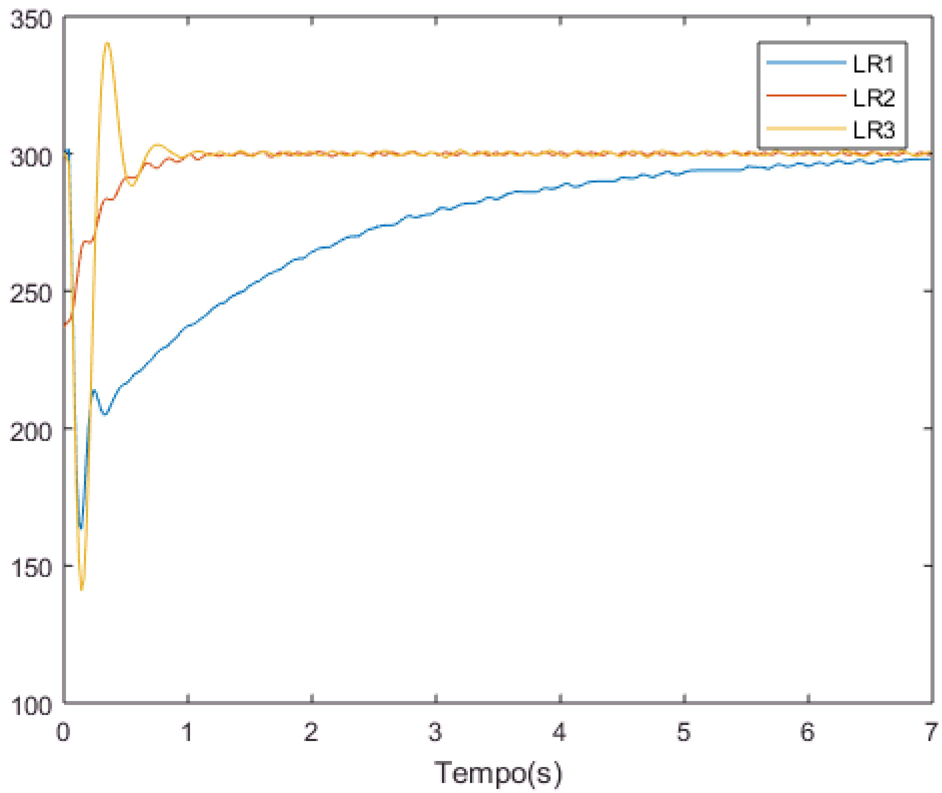
Escolha os ganhos K_p para os 3 casos para obter respostas rápidas e com pouca sobreelevação.

Depois, defina o valor do ganho Kp abaixo para cada escolha correspondente do zero e veja como ficam as respostas.

```
Ki=-pl/10; % primeiro LR
Kp=0.486;   % substituir Kp
[y1,u1,t1] = arduino_controle_PI_rc(obj,Ref,Tempo, Kp, Ki);
[y,u,t] = arduino_controle_PI_rc(obj,Ref,1, Kp, Ki);

Ki=-pl/2; % segundo LR
Kp=1;   % substituir Kp
[y2,u2,t2] = arduino_controle_PI_rc(obj,Ref,Tempo, Kp, Ki);
[y,u,t] = arduino_controle_PI_rc(obj,Ref,1, Kp, Ki);

Ki=-2*pl; % terceiro LR
Kp=0.25;   % substituir Kp
[y3,u3,t3] = arduino_controle_PI_rc(obj,Ref,Tempo, Kp, Ki);
figure
plot(t1,y1,t2,y2,t3,y3);
xlabel('Tempo(s)');
legend('LR1','LR2','LR3');
```



4.1 Justifique o efeito da escolha da localização do zero nas respostas obtidas, e sugira a melhor localização, comparando com o zero introduzido pela sintonia lambda.

A melhor localização dos zeros foi obtida no segundo LR, visto que a resposta foi a mais rápida, esse fenomeno acontece visto que afastamos o polo do 0 do eixo.