

## Aula 8 - Laboratório de Controle - 2022/1

### Modelagem e controle usando microcontrolador

Nomes: Arthur Macedo e Catarina Sastre

#### Atividade 0

Identificar porta serial do Arduino e testar resposta ao degrau com função `arduino_coleta()`.

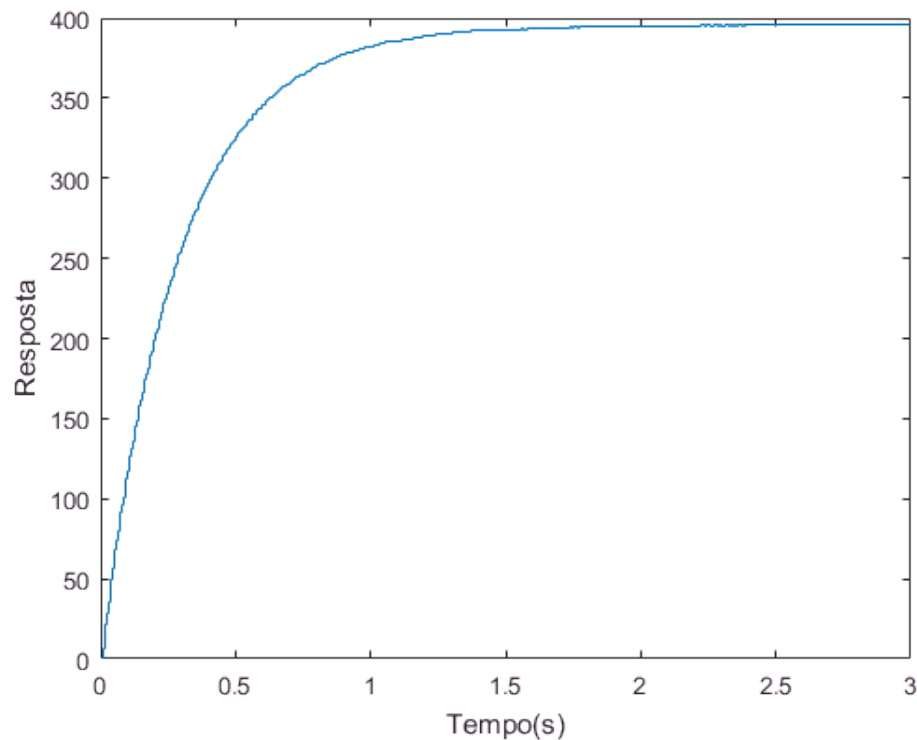
```
z=seriallist;  
comPort=z{3};  
obj=serial(comPort,'BaudRate',9600);  
obj.Terminator='CR';  
fopen(obj);
```

#### Atividade 1

Dar degraus e coletar a resposta usando o Arduino escolhendo Ref, Tempo,  $T_s=20$  (ms).

Dar degrau e obter ganho e constante de tempo, informando aqui.

```
zera_saida(obj);  
Ref=100;  
Ts=5;  
Tempo=3;  
[y,t] = arduino_coleta(obj,Ref,Ts,Tempo);  
figure  
stairs(t,y);  
xlabel('Tempo(s)');  
ylabel('Resposta');
```



```
j = sum(y<0.63*y(end));
tau = t(j)           % constante de tempo
```

```
tau = 0.2850
```

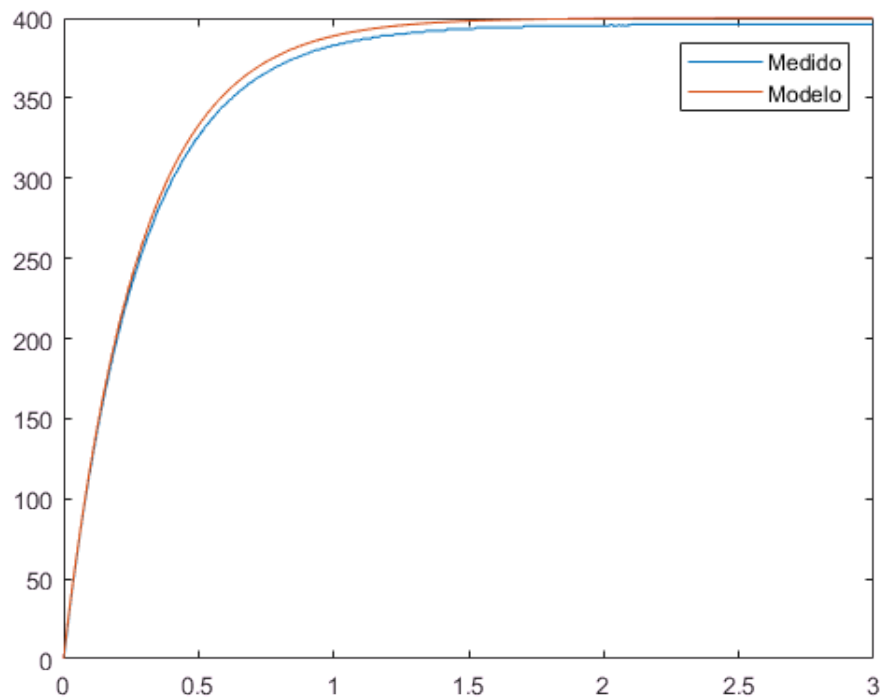
Qual a constante de tempo e ganho deste sistema?

Constante de tempo  $\tau = 0.28$  e o ganho do sistema é a saída (em regime) em relação à referência, sendo a referência = 100 e a saída = 400, o ganho do sistema foi 4.

## Atividade 2

Usar este ambiente para validar o modelo  $G(s) = \frac{K}{\tau s + 1}$  com pelo menos 3 novos degraus

```
K=4;
tau=0.28;
gl=tf(K,[tau 1]);
zera_saida(obj);
[y,t] = arduino_coleta(obj,Ref,Ts,Tempo);
ys=step(Ref*gl,t);
figure
plot(t,y,t,ys);legend('Medido','Modelo');
```



2.1 Comente a qualidade do modelo obtido, justificando.

Foi observado <sup>ordem 1</sup> que o modelo funciona com uma qualidade muito boa. Isso era esperado pois o sistema se trata **de ordem 0**, da mesma forma como o modelo. Dessa forma o modelo obtido teve resultados bem próximos do medido.

2.2 Compare e justifique a diferença do sinal de saída medido e simulada em regime, justificando.

Como pode ser visto abaixo, existe uma pequena diferença entre o sinal simulado e o medido. Essa diferença está atribuída ao erro de discretização por parte do arduino, pois estão sendo usados apenas 10 bits **para discretizar um sinal entre 0 a 5V**. <sup>Que erro isto produz?</sup>

```
dif = y(end) - ys(end)
```

```
dif = -3.9909
```

### Atividade 3

Projetar um controlador PI via método lambda de modo a ter constante de tempo de malha fechada igual à de malha aberta. Analisar a saída e o sinal de controle.

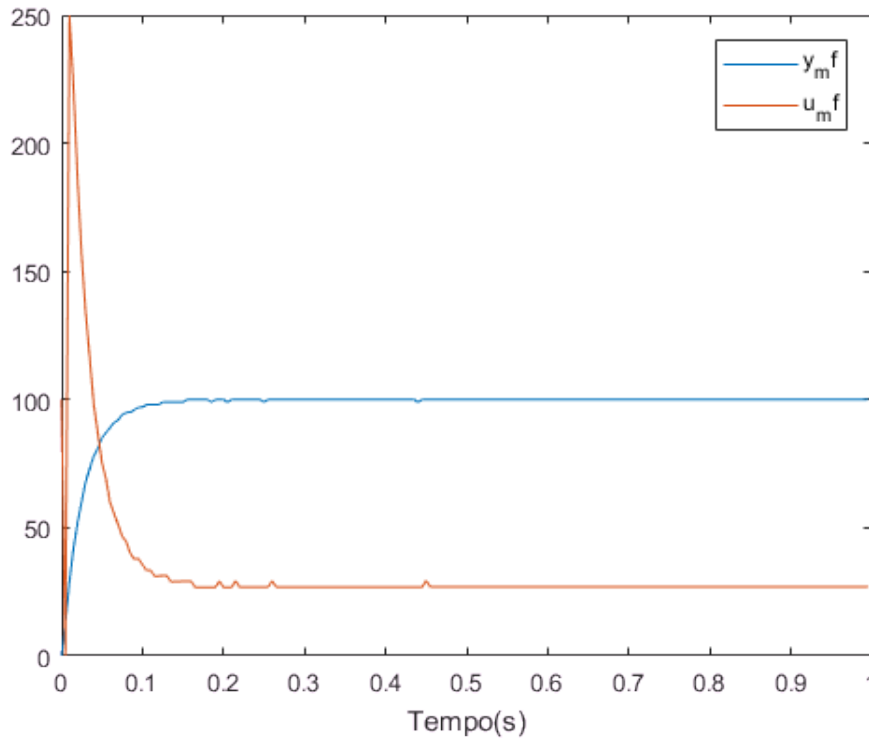
$$G_p = \frac{K}{\tau s + 1} K_p = \frac{\tau}{K\lambda} T_i = \frac{1}{K_i} = \tau C(s) = K_p + \frac{K_p K_i}{s}$$

```
lambda=0.1*tau;  
Kp=tau/(K*lambda);  
Ki=1/tau;
```

```

zera_saida(obj);
[y_mf,u_mf,t_mf] = arduino_controle(obj,Ref,Ts,1, floor(Kp*100), floor(100*Kp*Ki));
figure
plot(t_mf,y_mf,t_mf,u_mf);legend('y_mf','u_mf');
xlabel('Tempo(s)');

```



### 3.1 Justificar a escolha de lambda e compare a constante de tempo de malha aberta e malha fechada

Após vários testes com valores de lambda, definimos um valor tal que o sinal de controle  $u$  não extrapole o limite de 255 e ao mesmo tempo faça o sistema ter uma resposta mais rápida.

```

j_mf = sum(y_mf < 0.63*y_mf(end));
tau_mf = t_mf(j_mf)           % constante de tempo de malha fechada

tau_mf = 0.0250

```

A constante de tempo de malha fechada é 0.025 (0.08 - 0.055) enquanto a constante de tempo de malha aberta é 0.28. Isso ocorre pois em malha fechada definimos um valor de lambda que faz o sistema se estabilizar muito mais rápido, causando uma diminuição drástica na constante de tempo (instante em que a resposta atinge 63% do valor em regime).

### 3.2 Descreva o comportamento do sinal de controle e sua proximidade aos limites de sua saturação.

Inicialmente o sinal de controle tem um pico em 250, portanto não satura em 255. Este pico acontece pois para que o sistema seja mais rápido na subida, o sinal de controle deve ser alto inicialmente e diminua aos poucos enquanto a saída estabiliza no set-point.

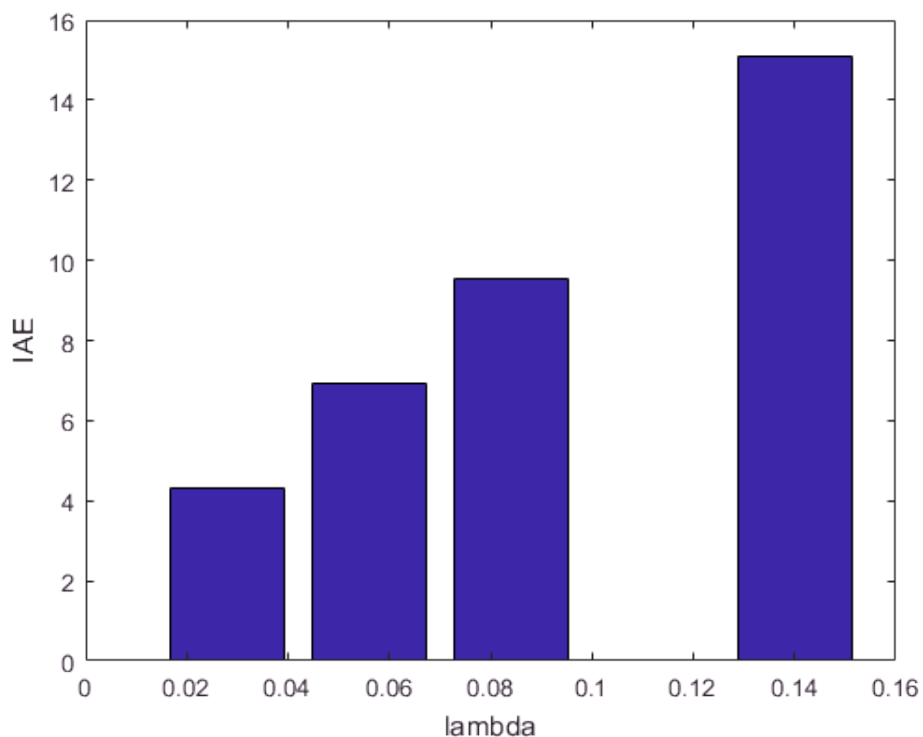
#### Atividade 4

Reduzir lambda para obter o IAE mínimo. Fazer um gráfico mostrando a relação de lambda com IAE mínimo.

Mostrar a resposta para o IAE mínimo.

Comparar o sinal de controle desta atividade com o da atividade 3.

```
lambda_2=[0.5 0.3 0.2 0.1]*tau;
for i=1:4
    Kp=tau/(K*lambda_2(i));
    Ki=1/tau;
    zera_saida(obj);
    [y_2,u_2,t_2] = arduino_controle(obj,Ref,Ts,Tempo, floor(Kp*100), floor(100*Kp*Ki));
    erro=Ref-y_2;
    iae(i,1)=trapz(t_2,abs(erro));
end
figure
bar(lambda_2,iae);
xlabel('lambda');ylabel('IAE');
```



4.1 Qual foi o valor mínimo de lambda? Por que não ficou menor?

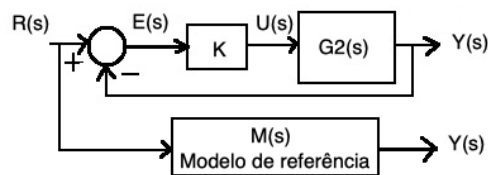
O valor mínimos de lambda é 0.028. Não é possível simular menor pois para lambda 10% de tau o sinal de controle tem um pico em 250, muito próximo do seu limite 255. Valores menores de lambda fazem o sinal de controle saturar e a simulação dá erro.

#### 4.2 Compare o sinal de controle para lambda mínimo e máximo

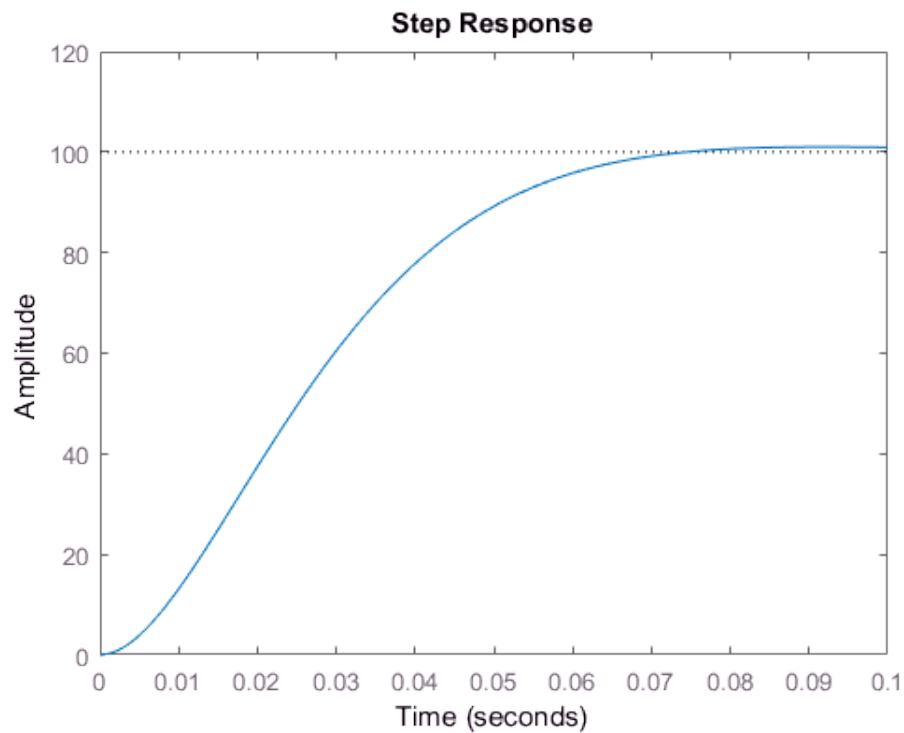
Para lambda mínimo o sinal de controle estabiliza o sistema mais rápido e, conseqüentemente, faz com que IAE tenha um valor menor. Lambda máximo, faz com que o pico do sinal de controle seja menor, estabilizando o sistema mais lentamente, o que aumenta o IAE.

#### Atividade 5:

A partir da melhor resposta da atividade 3, proponha um modelo de referência de segunda ordem  $M(s)$  tal que sua resposta seja semelhante à obtida na atividade 3. Para isto, meça a sobrelevação UP e o tempo de estabelecimento  $t_s$ .



```
UP=1;           % definimos 1% pois observamos que não há sobressinal na atividade 3.
ts=0.08;
a=log(UP/100);
zeta=sqrt(a^2/(pi^2+a^2));
wn=4/(ts*zeta);
m=tf(wn^2,[1 2*zeta*wn wn^2]);
figure
step(Ref*m)
```



5.1 Compare a resposta de  $M(s)$  com a obtida na atividade 3 que gerou UP e  $t_s$  utilizados.

No geral, a resposta do sistema ficou muito parecida com a resposta da atividade 3, principalmente se tratando de tempo de assentamento, sobressinal e valor em regime. É possível observar algumas pequenas características transitórias um pouco diferentes pois dessa vez se trata de um sistema de ordem 2.

O sistema original + PI também é de ordem 2!